

식단 관리를 통한

다이어트 플래너

Diet Planner

Only for you

By Danzi



Contents



01

프로젝트 개요

02

개발 기획

03

개발 과정

04

프로젝트 결과물 소개

05

QnA

Team Danzi



허동우(PM)

BE 개발
양추정 모델 설계



김유리

Cross Platform 개발
양추정 모델 설계



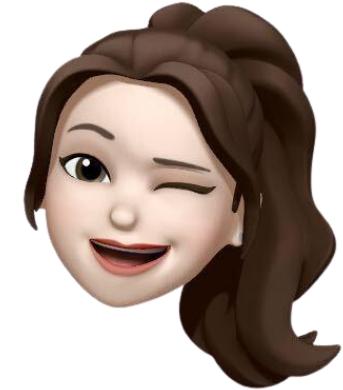
류가연

BE 개발
음식분류 모델 설계



송대선

Cross Platform 개발
데이터 정제



정지윤

객체탐지 모델 설계
음식분류 모델 설계

01

「프로젝트 개요」

By Danzi



> 의료뉴스 > 일반 > 헤드라인

국내 성인 비만율 40% 육박..남성 2명 중 1명은 비만

배준열 기자 | 승인 2024.02.15 10:43 | 댓글 0

대한비만학회 2023 비만 팩트시트 '숫자로 보는 비만' 발간
최근 10년간 성인 남녀 모두 증가..모든 연령 비만·복부비만 증가
폭음, 고열량 안주, 야식·배달음식 등 주원인..운동·약물요법 등 필요

"수박을 어떻게 참아요"...여름 당뇨환자 '혈당 다이어트' 적은

중앙일보 원문 | 기사전송 2024-06-29 05:01 | 최종수정 2024-06-29 06:34

3 3

AI챗으로 요약
↑ ⏴ ⏵

홈 > 경제속보 > 생활/문화

출시 한 달 안 돼 100만 개 이상 팔려..저당 멜론바 인기

이재욱 기자 | 입력 2024.06.30 00:35 | 댓글 0

컴포즈커피, 여름 한정 저당·저칼로리 음료 3종 출시

N 구독 | 남궁민관 기자 | 2024.06.20 13:51:52



제로슈거 파우더·사이다로 당·칼로리 최소화
캐모마일 리프레시·청포도 그린티·복숭아 아이스티 구성

푸드

제로 칼로리·저당 식품 열풍… 대체당 '알룰로스' 뛴다

신은진 헬스조선 기자
입력 2024/06/26 09:41

유통 > 식품

'저칼로리' 힘주는 농심, 누들핏 짜파구리·카구리 출시

농심, 저칼로리 제품 누들핏도 짜파구리·카구리 맛 출시
다이어트·식단 관리 중시하는 MZ타깃
4월 출시한 짜파게티 건면, 출시 1달 만 700만봉 팔려

How ?

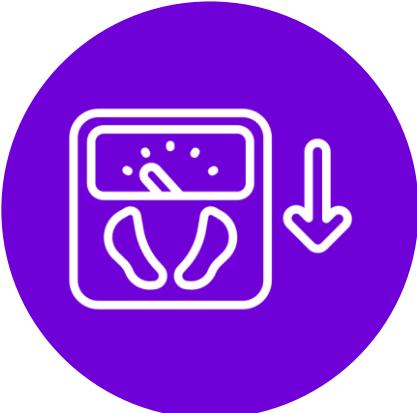
IF



일일 총 에너지 소비량



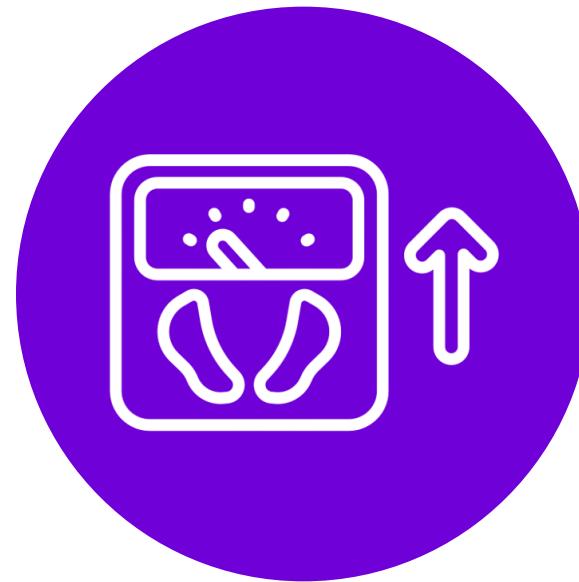
섭취 칼로리



체중 감소

How ?

ELSE :

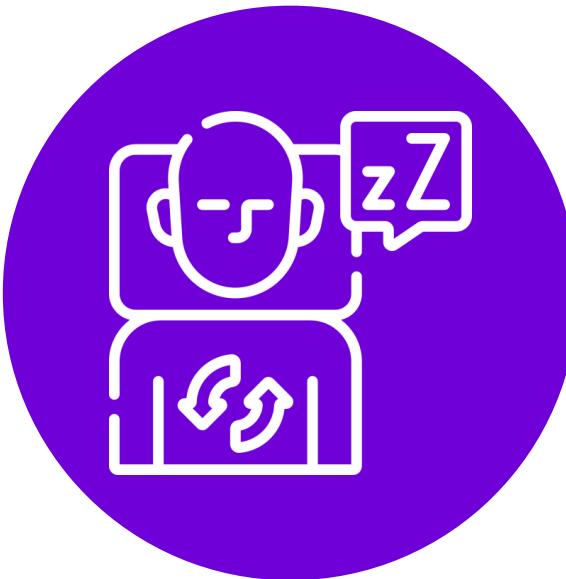


체중 증가

TDEE ?



=



X

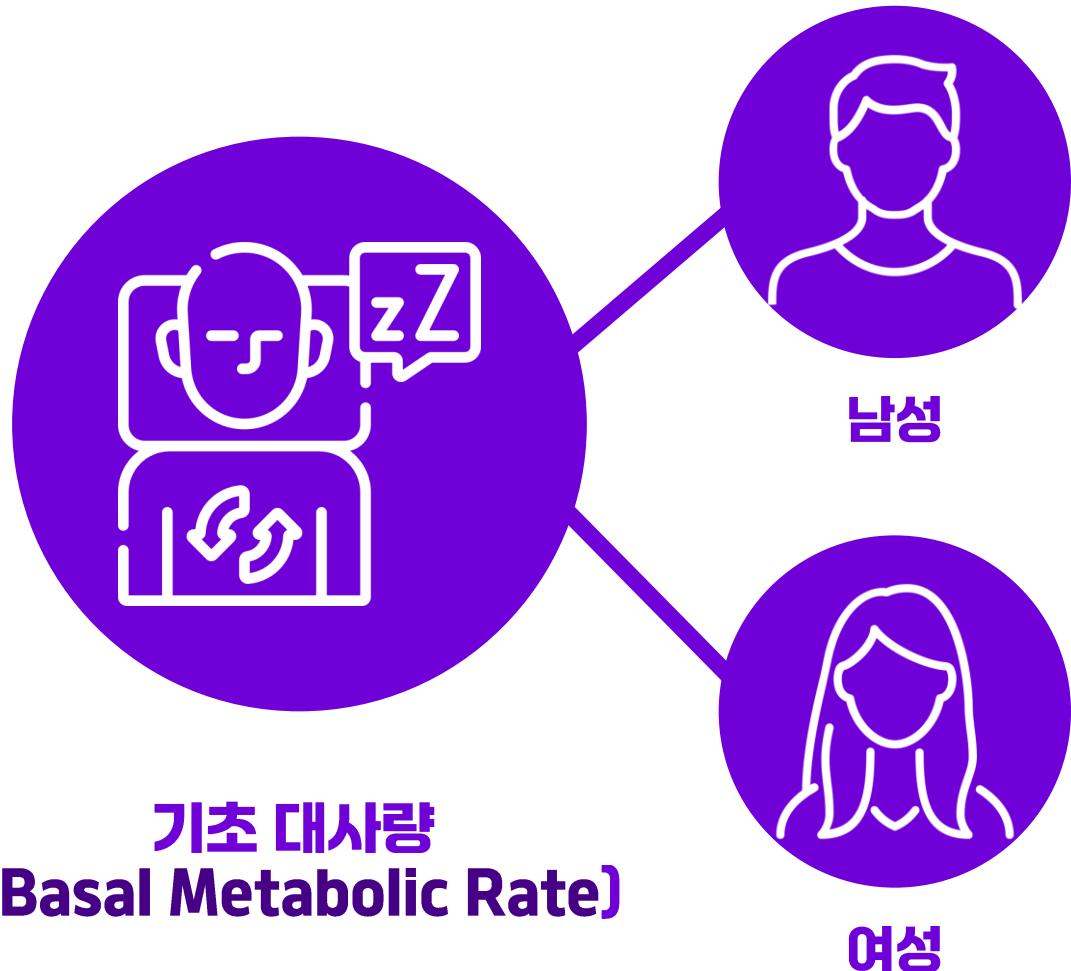


일일 총 에너지 소비량
[Total Daily Energy Expenditure]

기초 대사량
(BMR)

활동 계수
(AF)

BMR ?

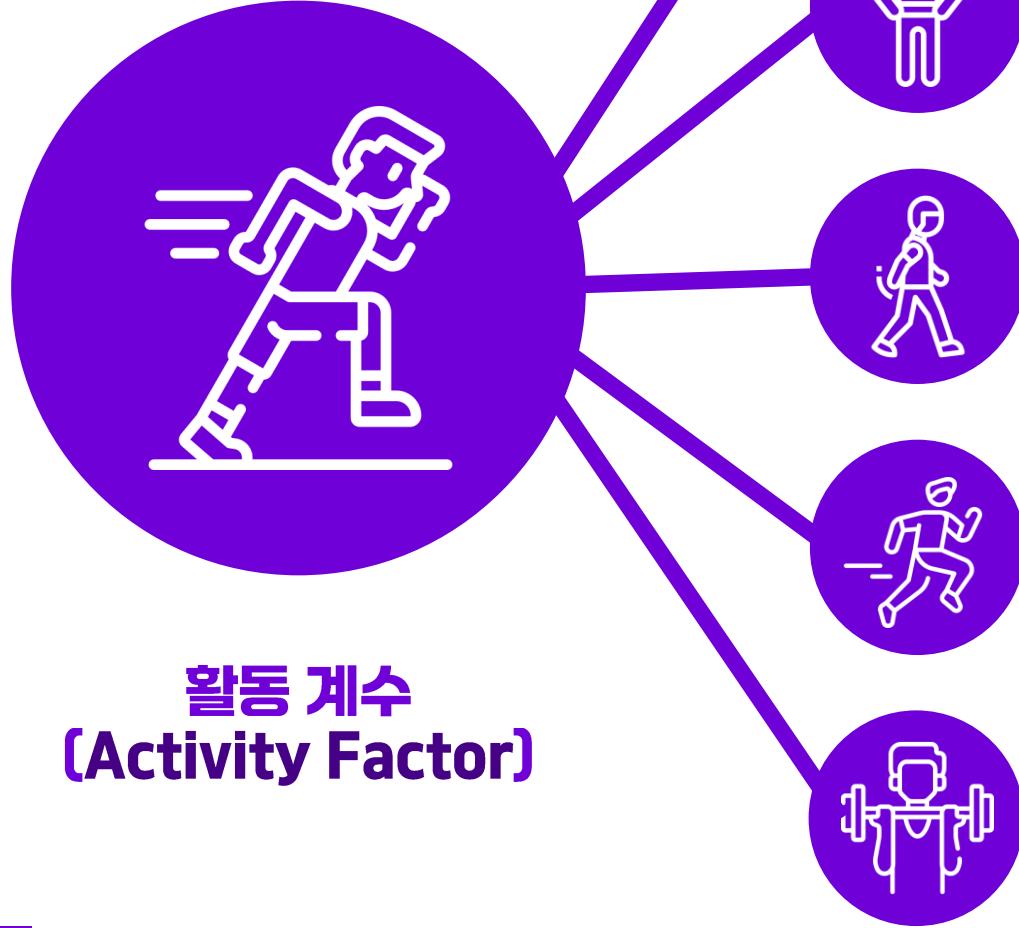


Harris Benedict 방정식

$$66.5 + (13.75 \times \text{체중 kg}) + (5.003 \times \text{키 cm}) - (6.75 \times \text{나이})$$

$$655.1 + (9.563 \times \text{체중 kg}) + (1.850 \times \text{키 cm}) - (4.676 \times \text{나이})$$

AF ?



운동을 거의 또는 전혀 하지 않음

$AF = 1.2$

적은 운동량 (주 1~3일의 스포츠 활동)

$AF = 1.375$

보통 운동량 (주 3~5일의 스포츠 활동)

$AF = 1.55$

많은 운동량 (주 6~7일의 스포츠 활동)

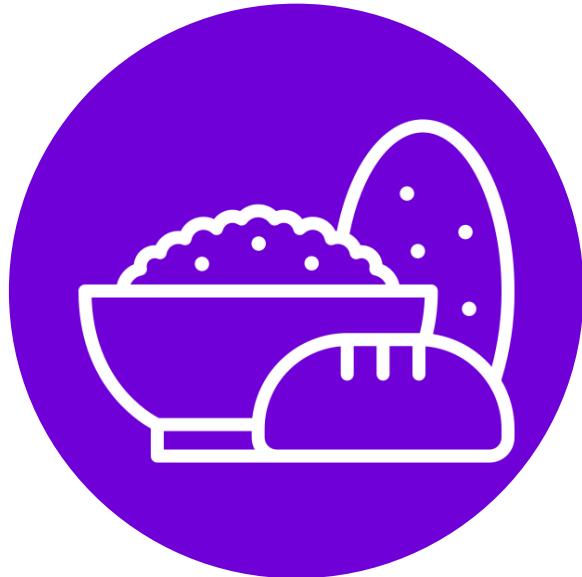
$AF = 1.725$

매우 많은 운동량 (매우 격렬한 스포츠, 운동선수)

$AF = 1.9$

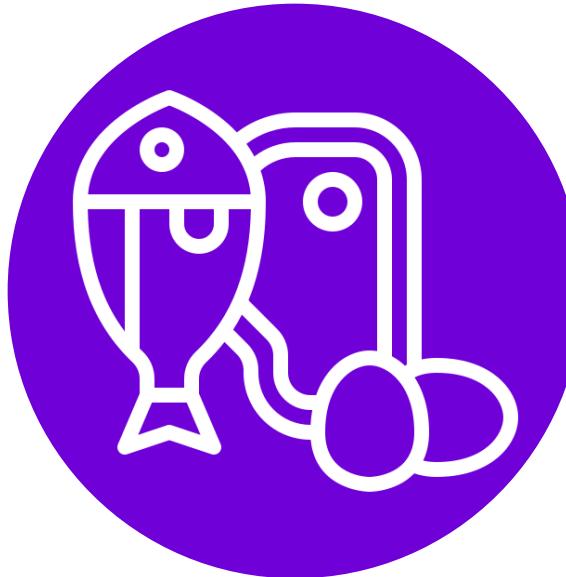
탄단지 ?

2020 한국인 영양소 섭취기준



탄수화물

55 ~ 65 %



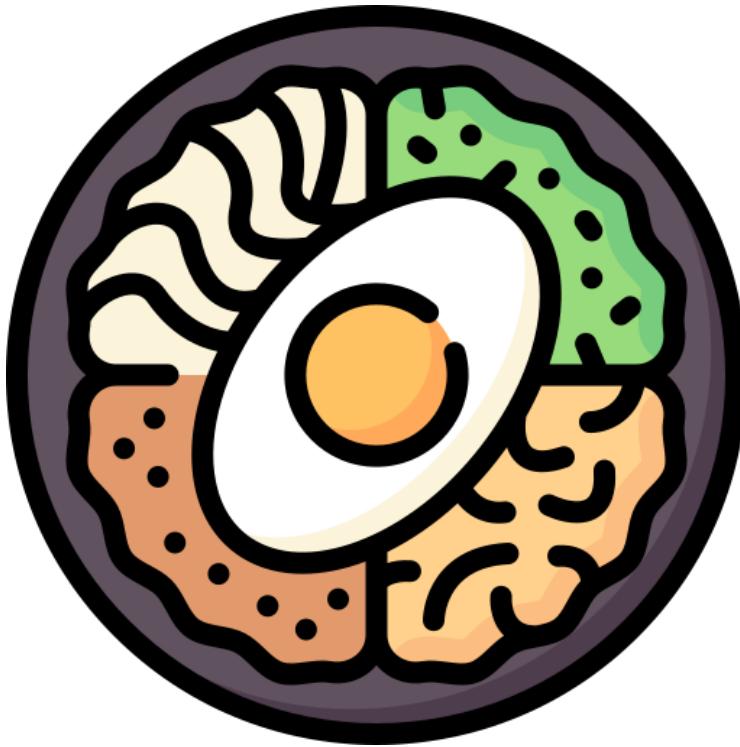
단백질

7 ~ 20 %



지방

15 ~ 30 %



음식이름 : 비빔밥

무 게 : 400 g

칼로리 : 586 kcal

탄수화물 : 89 g

단백질 : 22 g

지방 : 14 g

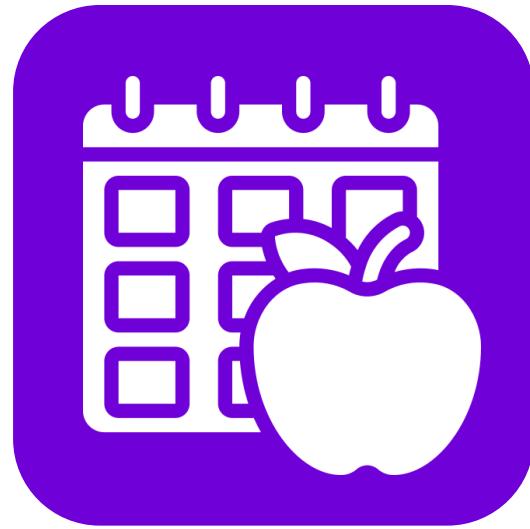
2024 7 JULY

SUN	MON	TUE	WED	THU	FRI	SAT
30	1 	2 	3 	4 	5 	6
7 	8 	9 	10 	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3



IMG to Kcal

음식 사진으로 부터
음식의 칼로리,
탄단지 산출



다이어트 캘린더

매일 섭취하는
음식의 사진을 통해
나만의 다이어트
캘린더 관리

On:ly

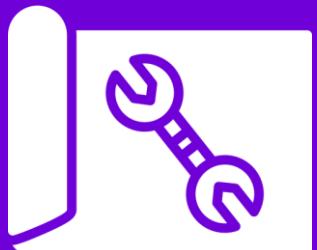
DIET PLANNER

ON:LY FOR YOU

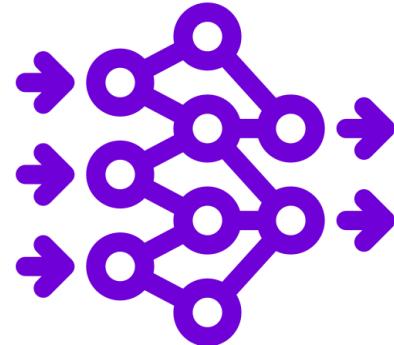
02

개발 기획

By Danzi



프로젝트 과정



1. 프로젝트 기획

주제 선정 및 기획

기능, 데이터 정의 등
표준문서 작성

2. 모델 설계&학습

데이터 구축 및 모델 설계

객체탐지, 음식분류, 양추정
모델 설계 및 학습

3. FE&BE 개발

APP 및 BE 서버 개발

모델&DB 서빙용 서버 구축
스마트폰 APP 개발

4. 테스트&시연

시스템 테스트 및 마무리

개발한 서비스의 테스트
발표 및 시연

일정 관리

요구사항 정의

대분류	요구사항	기능
MEM	회원가입	회원가입 기능
	로그인	로그인 기능
	로그아웃	로그아웃 기능
	회원탈퇴	회원탈퇴 기능
	회원정보수정	회원정보수정 기능
DIET	다이어트 계획수립	신체정보 입력
		다이어트 기간 설정
		하루 권장 섭취량 산출

대분류	요구사항	기능
DIET	다이어트 일정관리	다이어트 캘린더 생성
		식단 기록(카메라)
		식단 기록(갤러리)
		식단 기록(텍스트)
		식단 기록 수정
		식단 기록 삭제
		다이어트 진행상황 관리
DIET	다이어트 시각화	일일 칼로리 섭취량 추이 그래프
		다이어트 진행률 그래프
	다이어트 정보수정	신체정보 수정
		다이어트 기간 수정

기능 정의

기능	세부기능	APP 페이지
APP 초기 관리기능	회원가입	회원정보 입력 화면
	로그인	로그인 화면
사전 데이터 처리 기능	다이어트 정보 입력	신체정보 입력 화면
	하루 권장 섭취량 산출	다이어트 기간 입력 화면
메인 홈 기능	다이어트 목표일 표시	메인 홈 화면
	응원 메시지 표시	
	현재 & 목표 체중 표시	
	당일 섭취 영양소 표시	

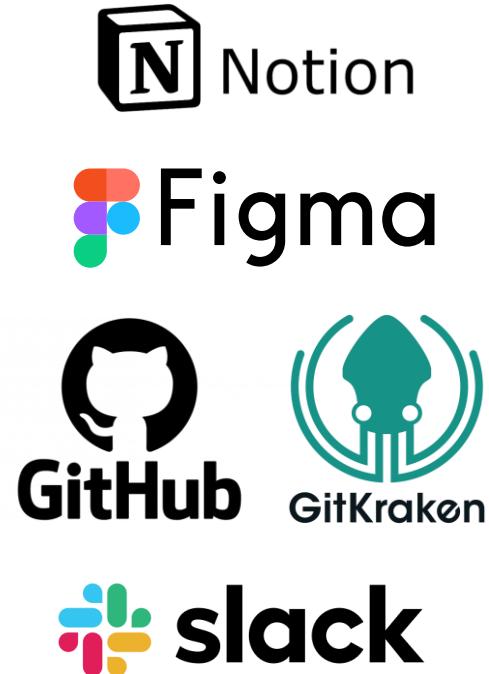
기능	세부기능	APP 페이지
식단 기록	식단 목록(갤러리)	식단 목록 화면_Empty
	식단 목록(갤러리)	식단 목록 화면_Selected
	식단 기록 화면	식단 기록 화면
식단 기록(카메라)	식단 목록 화면_Empty	식단 목록 화면_Empty
	식단 목록 화면_Selected	식단 목록 화면_Selected
	식단 기록 화면	식단 기록 화면
식단 기록(텍스트)	식단 목록 화면_Empty	식단 목록 화면_Empty
	식단 목록 화면_Selected	식단 목록 화면_Selected
	식단 기록 화면	식단 기록 화면
진행상황 시각화	일일 칼로리 섭취량 추이 그래프	진행상황 화면
	다이어트 진행률 그래프	

데이터 정의

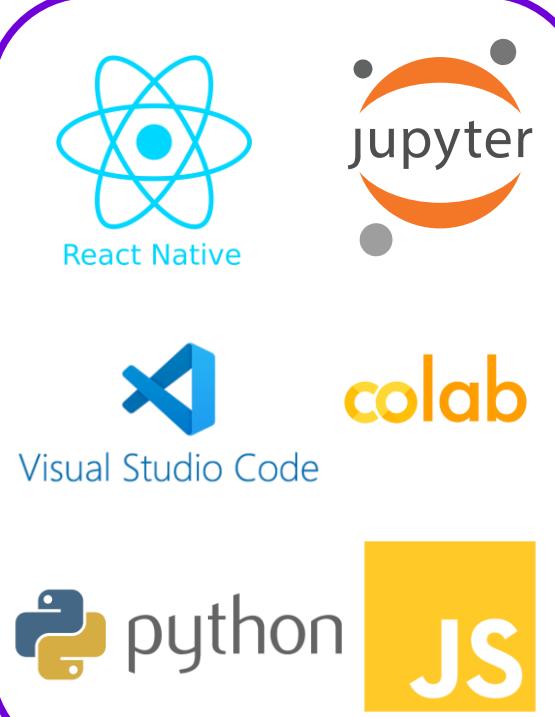
테이블	데이터	역할
user	사용자 ID, PW, 닉네임, 성별, 생년월일, 키, 체중, 회원가입 일자, 수정일자, 회원탈퇴 유무	회원 정보 관리
activity_level	활동 등급, 활동 등급 설명, 활동 계수	활동 계수 정보 제공
diet_period	다이어트 시작, 종료 일시, 다이어트 기간, 기초 대사량, 활동 대사량, 총 소비 칼로리, 일일 권장 칼로리, 탄수화물, 단백질, 지방	다이어트 기간 정보 관리, 다이어트 기간에 따른 필수 정보 제공
daily_diet	일일 식단 일자, 일일 다이어트 성공 여부, 일일 섭취 칼로리, 탄수화물, 단백질, 지방	일일 섭취 영양정보 관리, 일일 다이어트 성공 여부 제공
diet	음식명, 음식량, 칼로리, 탄수화물, 단백질, 지방, 식단 등록일자, 식단 수정일자, 식단 이미지, 식단 일자	사용자가 등록한 식단 정보 관리
food	음식명, 음식량, 칼로리, 탄수화물, 단백질, 지방	음식정보 테이블, 음식의 실제 영양정보 제공

Tech Stack

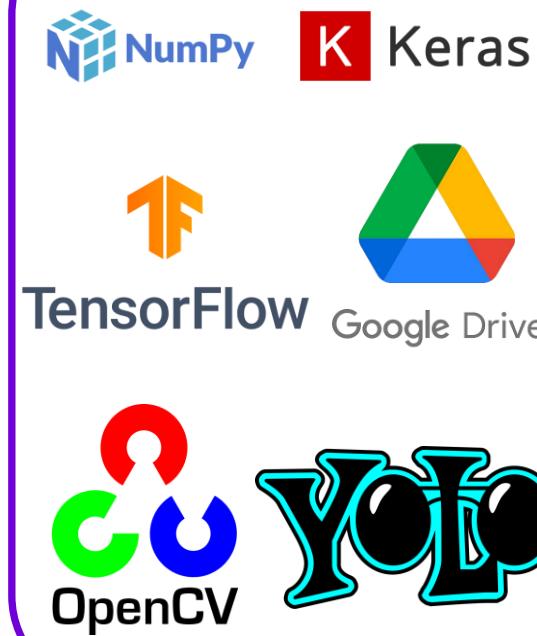
협업 및 버전관리



개발 언어 및 프레임워크



이미지 딥러닝



DB 및 서버



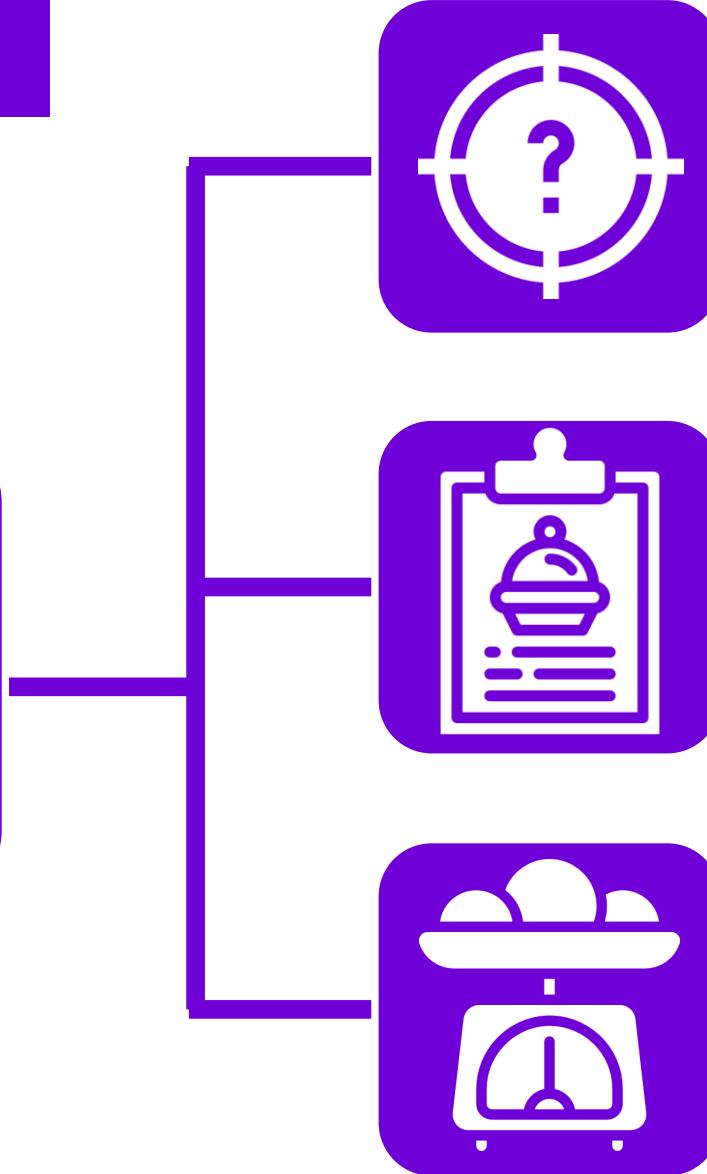
03

「개발 과정」

By Danzi



모델 개발



1. 객체 탐지 모델

이미지 속에서
음식과 동전을 탐지

2. 음식 분류 모델

이미지 속 음식이
어떤 음식인지 판별

3. 음식량 추정 모델

이미지 속 음식의
양을 1인분 기준으로
몇%에 해당하는지 판단

모델 개발



1. 객체 탐지 모델

이미지 속에서
음식과 동전을 탐지

The screenshot shows a user interface for extracting food images and nutritional information. At the top, there is a circular icon containing a meal (steak, egg, and vegetables). Below it, a heading reads "음식 이미지 및 영양정보 텍스트". A navigation bar includes categories like "#음식종류", "# 음식 양", "# 칼로리", "# 한식", and "# 이미지". Below the heading, there are tabs for "분야" (Food), "영상이미지" (Video Image), "유형" (Type), and "이미지" (Image). Further down, details are provided: "구축년도: 2020", "갱신년월: 2021-12", "조회수: 57,661", "다운로드: 6,577", and "용량: 1.60 TB". At the bottom, there are buttons for "다운로드" (Download) and "샘플 데이터" (Sample Data).

AI HUB. 음식 이미지 및 영양정보 텍스트

양추정 이미지 데이터셋
약 40,000장 (48GB) 사용

모델 개발



1. 객체 탐지 모델

이미지 속에서
음식과 동전을 탐지

김밥.jpeg



김밥.txt

```
0 0.768349 0.280041 0.088595 0.206212
1 0.890040 0.312882 0.061100 0.213849
2 0.635457 0.234980 0.071283 0.073320
3 0.426190 0.516802 0.662933 0.740835
4 0.431282 0.477088 0.404277 0.417006
```

데이터셋 내부의 사전 제공된
음식, 동전의 Bounding Box 정보를
그대로 활용

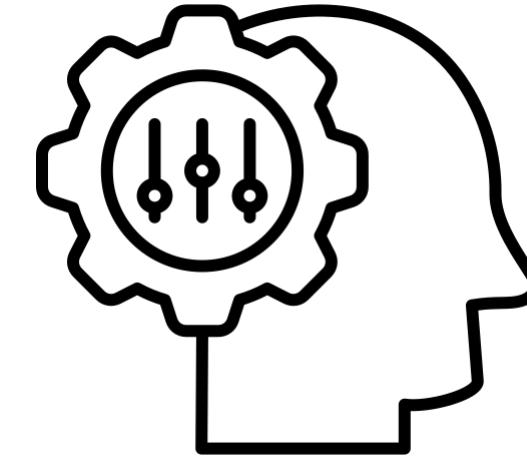
모델 개발



1. 객체 탐지 모델

이미지 속에서
음식과 동전을 탐지

Yolov3의 백본 신경망
Darknet53을 사용



darknet53.conv.74의
가중치를 기반으로
세부 사항을 변경하여 학습

모델 개발



1. 객체 탐지 모델

이미지 속에서
음식과 동전을 탐지

모델 번호	1	2	3
모델 정보	기본 모델 그대로 학습	val_size = 0.2 추가	배치사이즈 증가
Input Img	(416,416,3)	(416,416,3)	(416,416,3)
Learning File	Train_YoloV3.ipynb	Train_YoloV3.ipynb	Train_YoloV3.ipynb
Count Img	36196	36196	36196
batch	8	8	64
subdivisions	1	1	8
val_data 분리	X	0	0
Last_loss	0.299	0.193	0.120
Avg_loss	0.289	0.273	0.210

모델 개발



1. 객체 탐지 모델

이미지 속에서
음식과 동전을 탐지

Avg_loss값이

0.210으로 가장 낮은

3번 모델을 최종 모델로 사용

3	배치사이즈 증가
(416,416,3)	
ain_YoloV3.ipynb	
36196	
64	
8	
val_data 분리	x
Last_loss	0.299
Avg_loss	0.289
	0
	0.120
	0.210

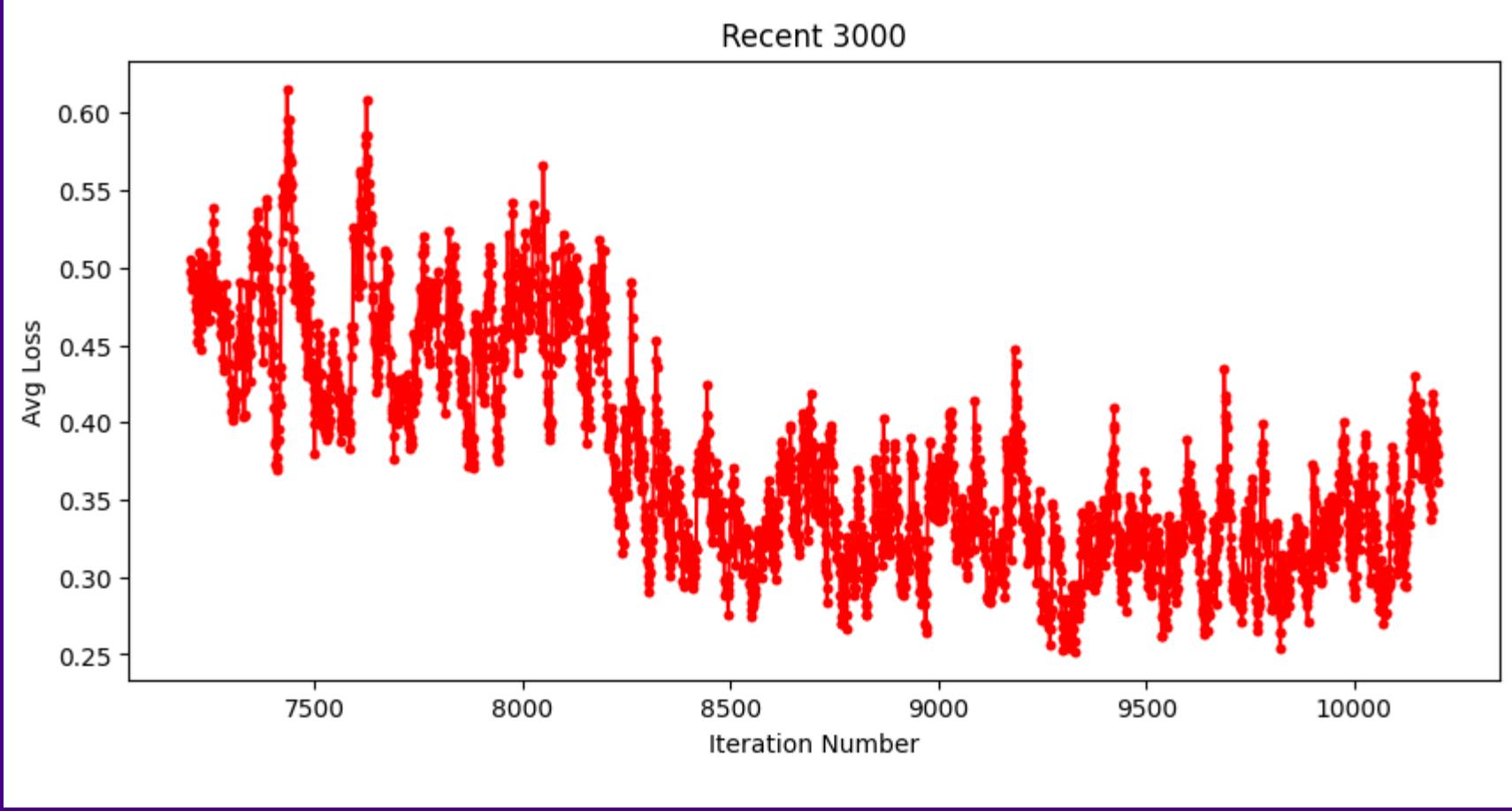
모델 개발

최종 모델 학습



1. 객체 탐지 모델

이미지 속에서
음식과 동전을 탐지



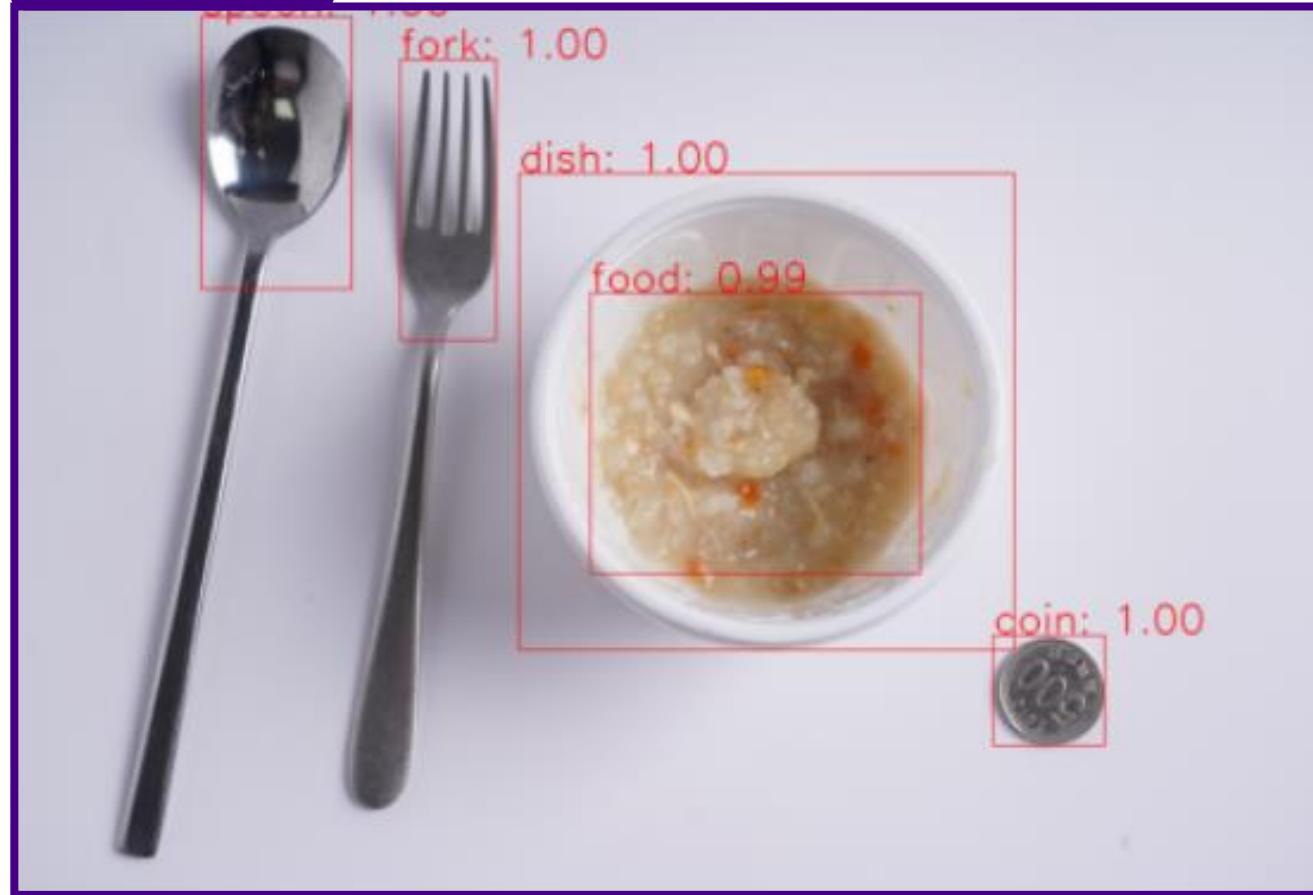
모델 개발



1. 객체 탐지 모델

이미지 속에서
음식과 동전을 탐지

객체 탐지 예시



모델 개발



2. 음식 분류 모델

이미지 속 음식이
어떤 음식인지 판별

The screenshot shows a card for a dataset titled "건강관리를 위한 음식 이미지" (Food images for health management). At the top right are several tags: #건강관리, #영양분석, #식이관리, #음식이미지, #음식정보, #접근성, and #사용성. Below the tags is a small icon of a fork and knife. The main title is "건강관리를 위한 음식 이미지". Underneath it, there are category labels: 분야 (Category), 헬스케어 (Healthcare), 유형 (Type), and 이미지 (Image). Below these are details: 구축년도: 2020, 개선년월: 2021-08, 조회수: 18,669, 다운로드: 4,212, 용량: 844.79 GB. At the bottom are two buttons: "다운로드" (Download) and "샘플 데이터" (Sample Data) with a question mark icon.

AI HUB. 건강관리를 위한 음식 이미지

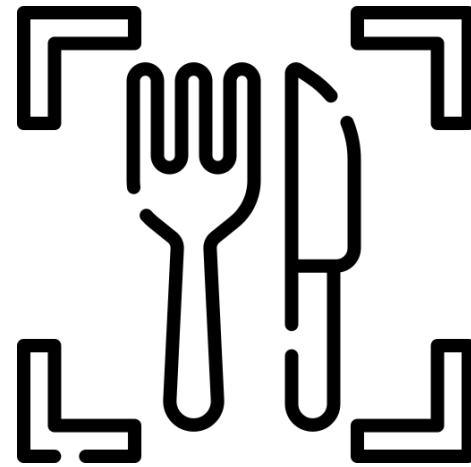
음식분류 이미지 데이터셋
약 100,000 장(188GB) 사용

모델 개발



2. 음식 분류 모델

이미지 속 음식이
어떤 음식인지 판별



X 101,643 장

약 500개의 음식 메뉴 중
106개의 음식을 학습에 활용

모델 개발



2. 음식 분류 모델

이미지 속 음식이
어떤 음식인지 판별

갈비구이.jpeg



갈비구이_cropped.jpeg



갈비구이.json

```
"Name": "galbigu-i",
"W": "0.999421296296296",
"H": "0.903310104529617",
"File Format": "jpg",
"Annotation Type": "binding",
"Point(x,y)": "0.500289351851852,0.548344947735192"
```

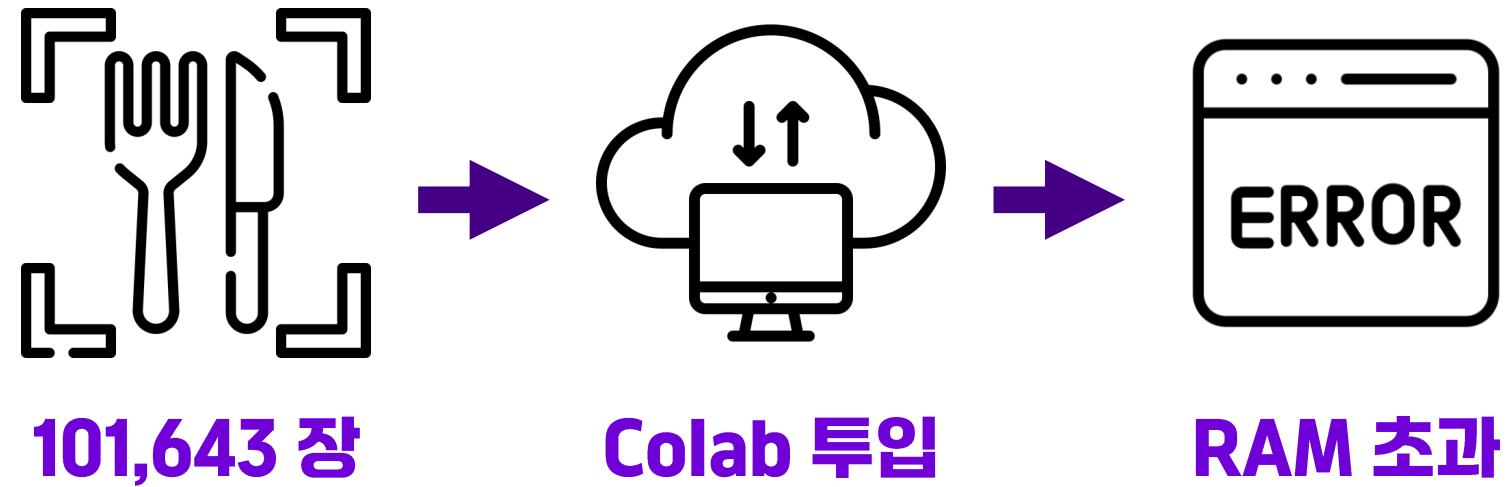
데이터셋 내부의 사전 제공된
음식의 Bounding Box 정보를
활용하여 음식 메뉴당
1000장을 crop

모델 개발



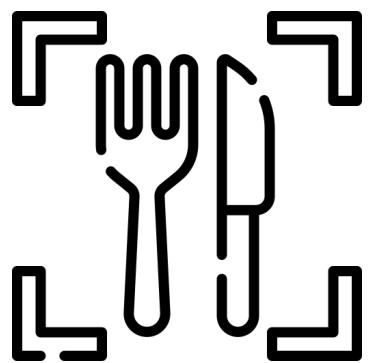
2. 음식 분류 모델

이미지 속 음식이
어떤 음식인지 판별

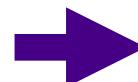


106개의 음식을
한번에 학습시키고자 했으나,
Colab의 시스템 RAM 용량을 넘김

모델 개발



101,643 장
(Main Set)

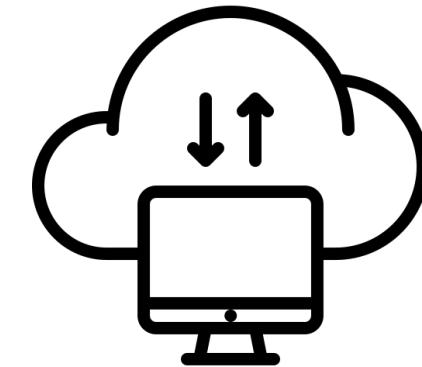
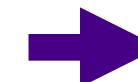


n 장
(Set 1)

...



n 장
(Set N)



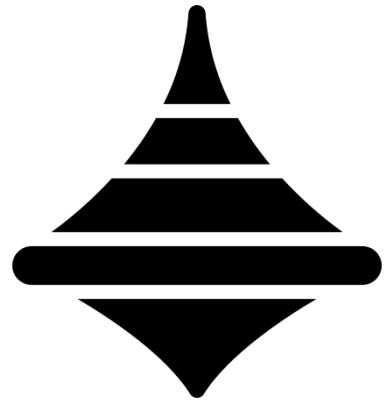
n 장씩
Colab 투입

음식당 1000개의 이미지 → 음식당 n개의 이미지

한번에 학습하는 이미지의 양을 줄여

RAM 할당량을 낮춤

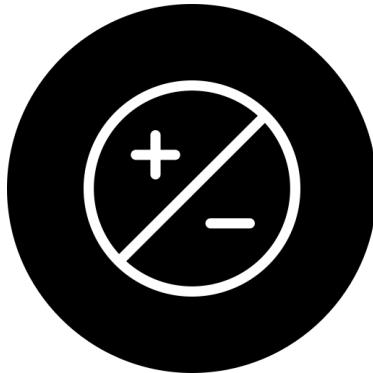
모델 개발



Inception V3

음식 분류 모델

논문 관련 높은 정확도



ResNet152V2

이미지 분류 모델

논문 관련 높은 정확도

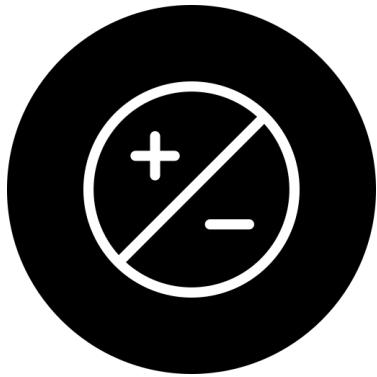


MobileNetV3small

모델 서빙을 고려한

경량화 된 모델

모델 개발



ResNet152V2

이미지 분류 모델

논문 관련 높은 정확도



MobileNetV3small

모델 서빙을 고려한

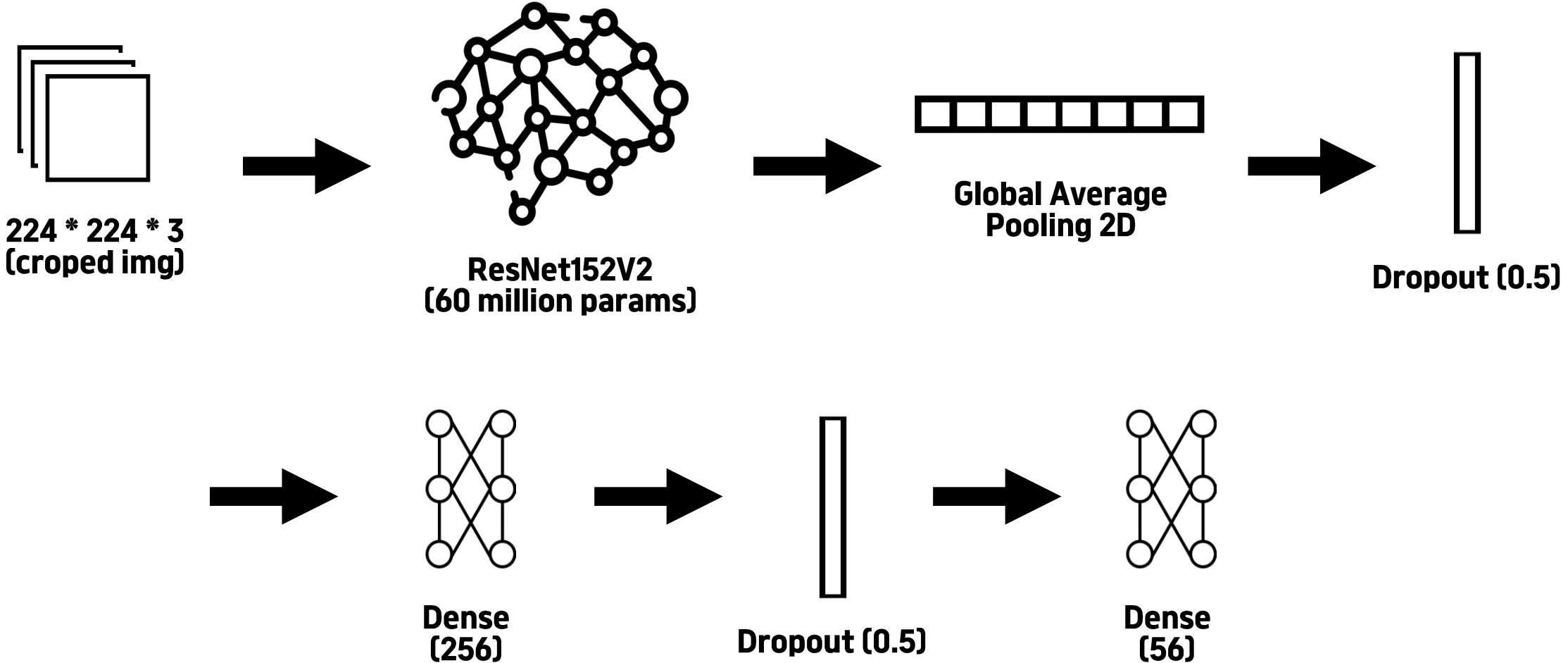
경량화 된 모델

전이학습에 사용할 모델 3가지를 선정

InceptionV3 모델은 학습 시간이 너무 오래 걸려 기각

모델 개발

ResNet152V2



모델 개발

ResNet152V2

모델번호	1	2	3	4	5
Input Img	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)
Model File	b1_14.hdf5	b1_47.hdf5	b2_10.hdf5	b3_10.hdf5	b4_05.hdf5
Epochs	100	100	100	100	100
Count Img	20000	20000	20000	20000	9998
Total params	58,869,042 (224.57MB)	58,869,042 (224.57MB)	58,869,042 (224.57MB)	58,869,042 (224.57MB)	58,869,042 (224.57MB)
Optimizer	ADAM	ADAM	ADAM	ADAM	ADAM
Val_Accuracy	0.8767	0.8831	0.9067	0.8942	0.9117
Val_loss	0.4379	0.4942	0.3456	0.3577	0.3214

모델 개발

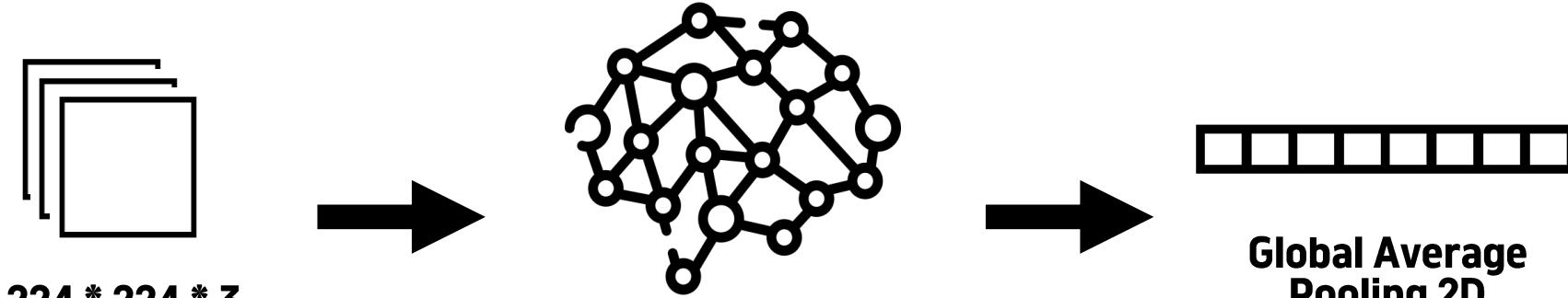
ResNet152V2

모델번호	1	2	3	4	5
Input Img					(224,224,3)
Model File					b4_05.hdf5
Epochs					100
Count Img					9998
Total params					58,869,042 (224.57MB)
Optimizer	ADAM	ADAM	ADAM	ADAM	ADAM
Val_Accuracy	0.8767	0.8831	0.9067	0.8942	0.9117
Val_loss	0.4379	0.4942	0.3456	0.3577	0.3214

ResNet152V2
최고 검증 정확도
91%

모델 개발

MobileNetV3small

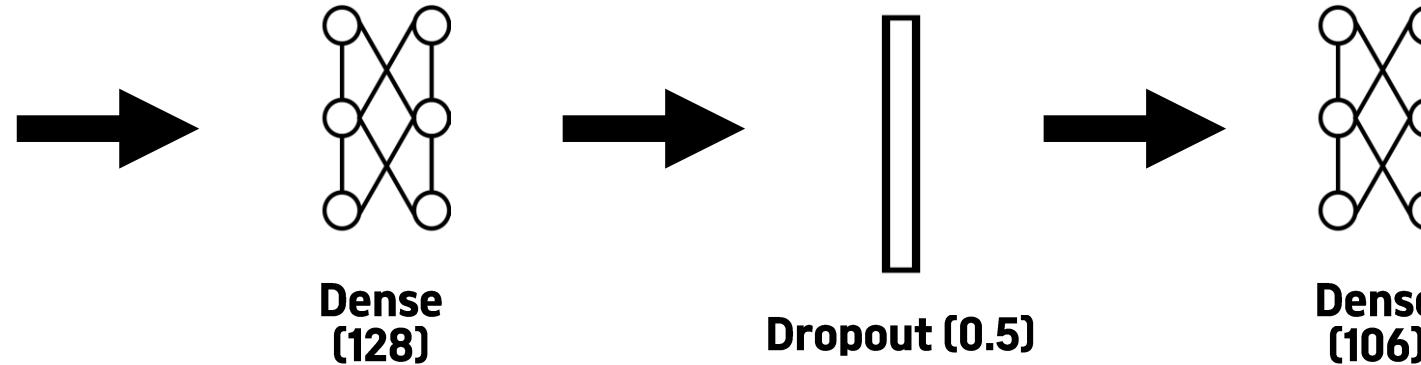


$224 * 224 * 3$
(cropped img)

MobileNetV3small
(2.5 million params)



Global Average
Pooling 2D



Dense
(128)

Dropout (0.5)

Dense
(106)

모델 개발

MobileNetV3small

모델번호	1	2	3	4	5	6	7
Input Img	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)
Model File	SGD_checkpoint_4.h5	SGD_checkpoint_5.h5	SGD_checkpoint_6.h5	SGD_checkpoint_7.h5	SGD_checkpoint_8.h5	SGD_checkpoint_9.h5	SGD_checkpoint_10.h5
Epochs	200	200	200	200	200	200	200
Count Img	13,568	13,568	13,568	6,667	13,568	13,568	13,568
Total params	1,026,650 (3.92 MB)						
Optimizer	SGD						
Val_Accuracy	0.9108	0.9046	0.9801	0.9933	0.9683	0.9827	0.9495
Val_loss	0.2516	0.2552	0.0648	0.0244	0.0953	0.0675	0.1391

모델 개발

MobileNetV3small

모델번호	1	2	3	4	5	6	7
Input Img	(224,224,3)					224,3)	(224,224,3)
Model File	SGD_checkpoint_4.h5					checkp_9.h5	SGD_checkpoint_10.h5
Epochs	200					00	200
Count Img	13,568					13,568	13,568
Total params	1,026,650 (3.92 MB)					6,650 (0.92 MB)	1,026,650 (3.92 MB)
Optimizer	SGD	SGD	SGD	SGD	SGD	SGD	SGD
Val_Accuracy	0.9108	0.9046	0.9801	0.9933	0.9683	0.9827	0.9495
Val_loss	0.2516	0.2552	0.0648	0.0244	0.0953	0.0675	0.1391

MobileNetV3small
최고 검증 정확도

99%

모델 개발

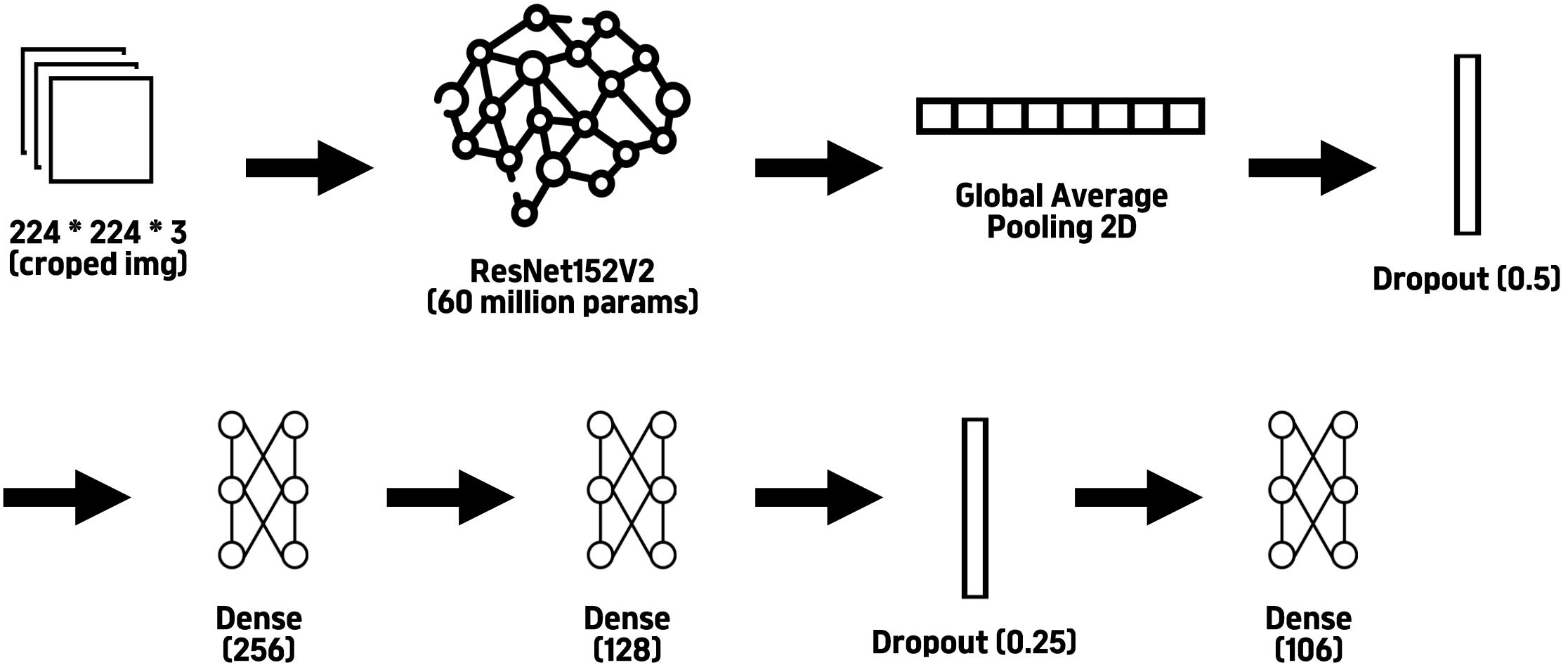
MobileNetV3small

모델번호	1	2	3	4	5	6	7
Input Img	(224,224,3)					224,3)	(224,224,3)
Model File	SGD_checkpoint_4.h5				checkp_9.h5	SGD_checkpoint_10.h5	
Epochs	200				00	200	
Count Img	13,568				568	13,568	
Total params	1,026,650 (3.92 MB)				6,650 (0.92 MB)	1,026,650 (3.92 MB)	
Optimizer	SGD	SGD	SGD	SGD	SGD	SGD	SGD
Val_Accuracy	0.9108	0.9046	0.9801	0.9933	0.9683	0.9827	0.9495
Val_loss	0.2516	0.2552	0.0648	0.0244	0.0953	0.0675	0.1391

But, 데이터셋을 나누는 과정에서
실수가 있었음
편향적인 결과가 도출되며
모델 기각

모델 개발

ResNet152V2_new



모델 개발

ResNet152V2_new

모델번호	1	2	3	4	5	6
Input Img	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)
Model File	nn1_100.hdf5	nn1_mode1.hdf5	nn2_mode1.hdf5	nn3_mode1.hdf5	nn4_mode1.hdf5	nnn_mode1.hdf5
Epochs	100	124	62	44	47	57
Count Img	20000	20000	20000	20000	20000	20000
Total params	58,947,684 (224.87MB)	58,947,684 (224.87MB)	58,947,684 (224.87MB)	58,947,684 (224.87MB)	58,947,684 (224.87MB)	58,947,684 (224.87MB)
Optimizer	SGD	SGD	SGD	SGD	SGD	SGD
Val_Accuracy	0.7467	0.7486	0.7636	0.7964	0.7947	0.7553
Val_loss	0.9005	0.8890	0.7840	0.7124	0.6943	0.8513

모델 개발

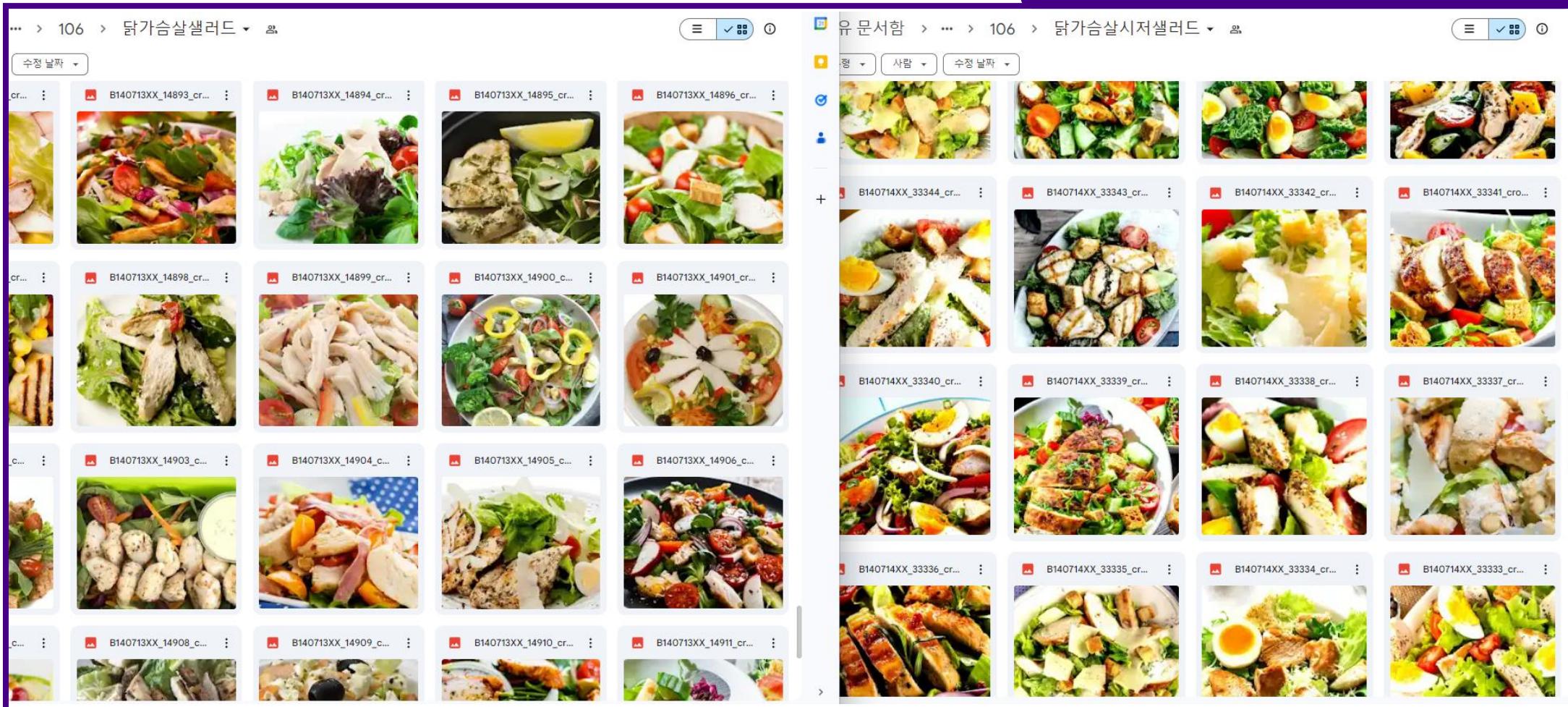
ResNet152V2_new

모델번호	1	2	3	4	5	6
Input Img						(224,224,3)
Model File						nnn_mode l.hdf5
Epochs						57
Count Img						20000
Total params						58,947,684 (224.87MB)
Optimizer	SGD	SGD	SGD	SGD	SGD	SGD
Val_Accuracy	0.7467	0.7486	0.7636	0.7964	0.7947	0.7553
Val_loss	0.9005	0.8890	0.7840	0.7124	0.6943	0.8513

모델 개발

ResNet152V2_new

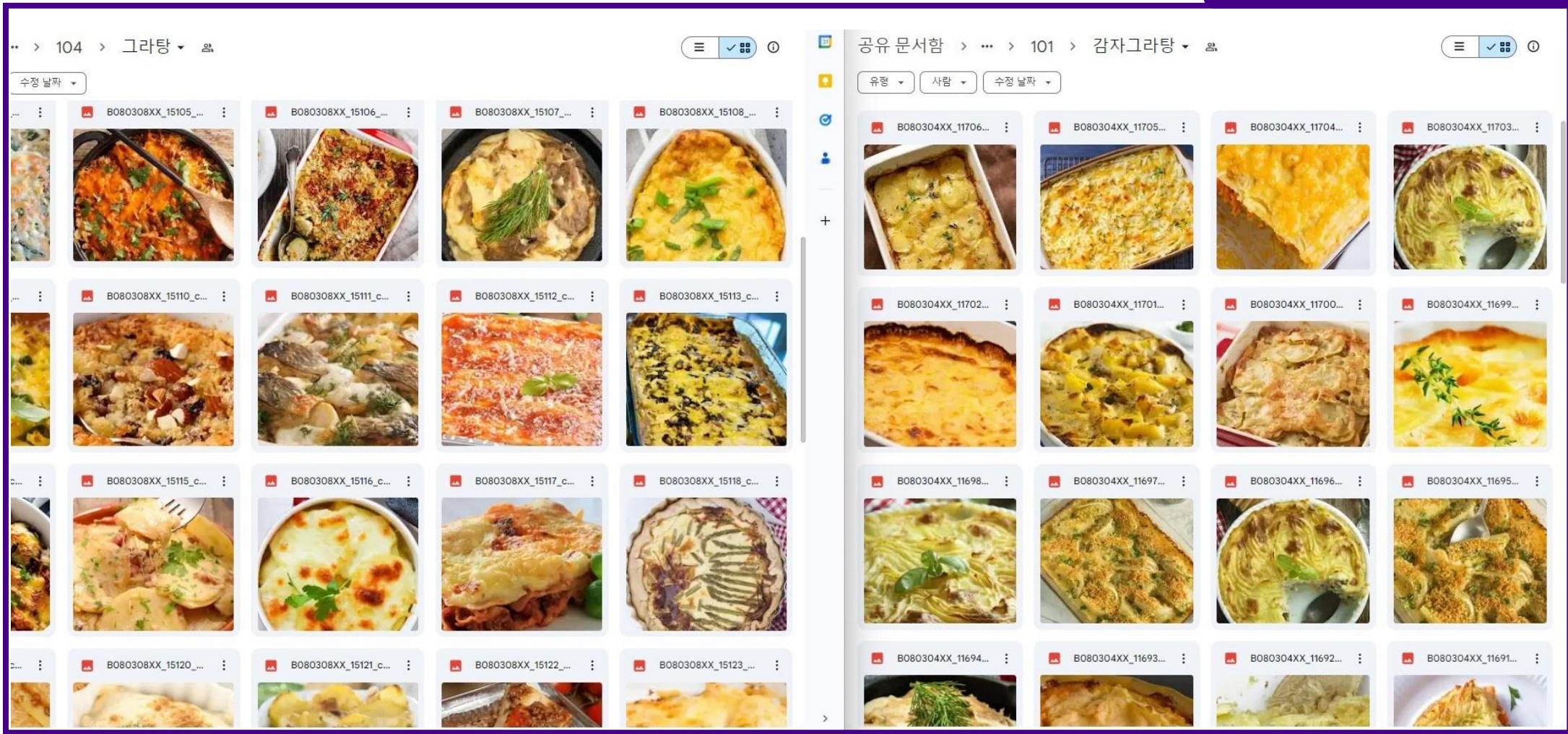
닭가슴살 샐러드 VS 닭가슴살 시저샐러드



모델 개발

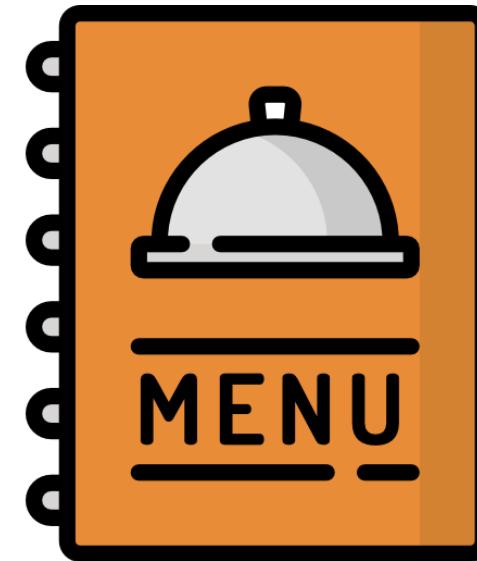
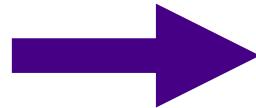
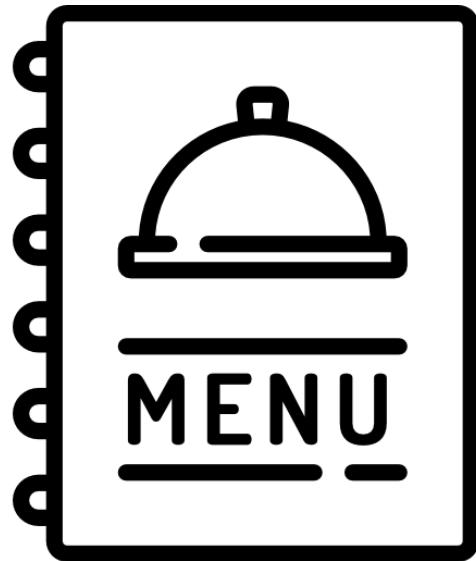
ResNet152V2_new

그라탕 VS 감자그라탕



모델 개발

ResNet152V2_new



106 가지 음식메뉴
(AI HUB에 등록된 순서대로)

100 가지 음식메뉴
(최대한 겹치는 음식 없게)

음료, 소스류, 외형이 비슷하거나 범주가 크게 겹치는 음식을 제외

팀 내부 회의를 통해 음식 100가지를 선출

모델 개발

ResNet152V2_new

모델번호	1	2	3	4	5
Input Img	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)
Model File	100_model8	add_model21	add_model6	add_model7	add_model13
Epochs	53	22	23	17	7
Count Img	20,000	20,000	20,000	19,996	20,000
Total params	59,545,956 (227.15MB)	59,545,956 (227.15MB)	59,545,956 (227.15MB)	59,545,956 (227.15MB)	59,545,956 (227.15MB)
Optimizer	ADAM	ADAM	ADAM	ADAM	ADAM
Val_Accuracy	0.7922	0.8453	0.8531	0.8719	0.8922
Val_loss	0.7485	0.5731	0.5182	0.4506	0.3806

모델 개발

ResNet152V2_new

모델번호	1	2	3	4	5
Input Img					(224,224,3)
Model File					add_model13
Epochs					7
Count Img					20,000
Total params					59,545,956 (227.15MB)
Optimizer	ADAM	ADAM	ADAM	ADAM	ADAM
Val_Accuracy	0.7922	0.8453	0.8531	0.8719	0.8922
Val_loss	0.7485	0.5731	0.5182	0.4506	0.3806

ResNet152V2_new

최고 검증 정확도 89%

5번 모델 최종 모델 결정

모델 개발



3. 음식량 추정 모델

이미지 속 음식의
양을 1인분 기준으로
몇%에 해당하는지 판단

#음식종류 #음식 양 #칼로리 #한식 #이미지

음식 이미지 및 영양정보 텍스트

분야 영상이미지 유형 이미지

구축년도: 2020 갱신년월: 2021-12 조회수: 57,661 다운로드: 6,577 용량: 1.60 TB

다운로드 샘플 데이터 ?

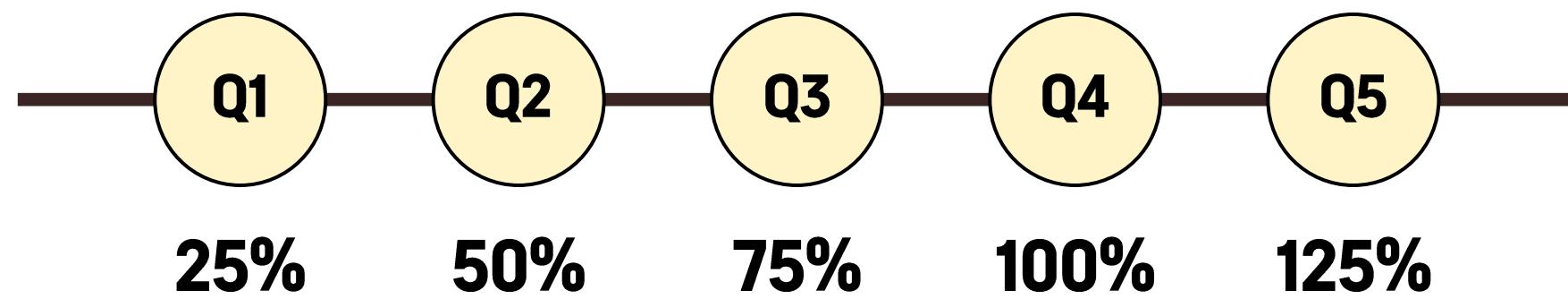
AI HUB. 음식 이미지 및 영양정보 텍스트

양추정 이미지 데이터셋
약 40,000장 (48GB) 사용

모델 개발



1인분 기준 Quantity

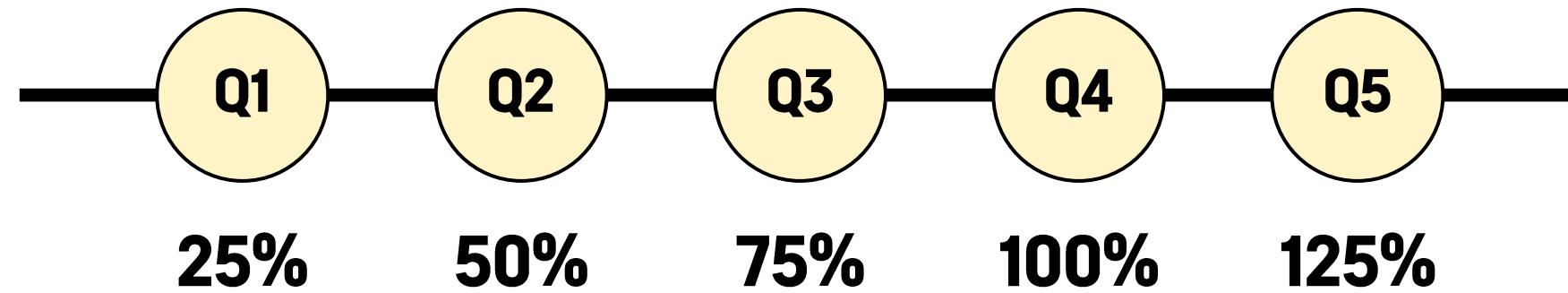


3. 음식량 추정 모델

이미지 속 음식의
양을 1인분 기준으로
몇%에 해당하는지 판단

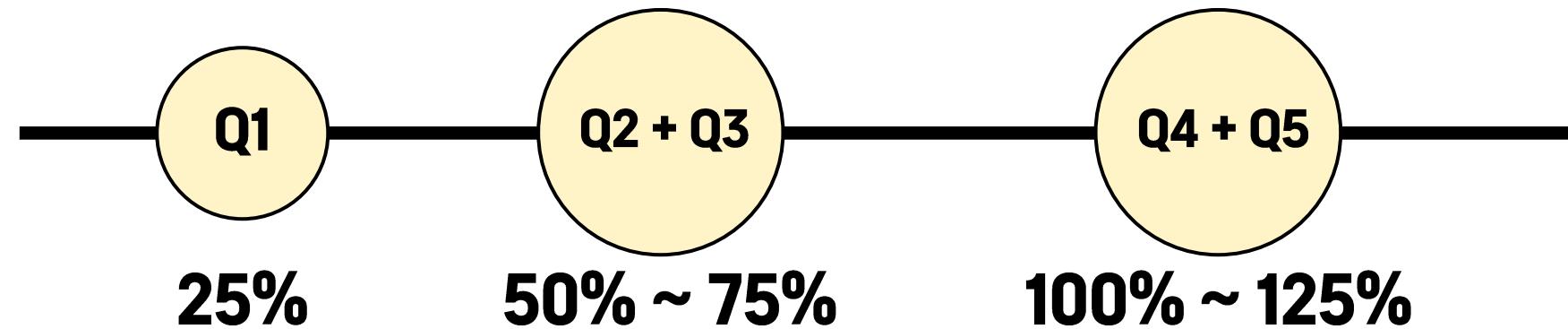
모델 개발

모델의 정확도가 높은 쪽은?



3. 음식량 추정 모델

이미지 속 음식의
양을 1인분 기준으로
몇%에 해당하는지 판단



모델 개발

모델의 정확도가 높은 쪽은?



3. 음식량 추정 모델

이미지 속 음식의
양을 1인분 기준으로
몇%에 해당하는지 판단

모델명	Q1~Q5 (1)	Q1~Q5 (2)	Q1, Q2+Q3, Q4+Q5 (1)	Q1, Q2+Q3, Q4+Q5 (2)
Input Img	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)
Learning File	model7.ipynb	model7-1.ipynb	model8.ipynb	model8.ipynb
Preprocess	crop_black	crop_black	crop_black	crop_black
Count Img	16600	16600	16600	16600
Epochs	100	100	100	100
전이학습 여부	전이학습	ResNet152 전이학습	ResNet152 전이학습	수제 모델
Total params	223.02MB	223.08MB	222.77MB	222.79MB
Trainale params	222.47MB	222.53MB	222.22MB	222.24MB
Val_Accuracy	0.5636	0.6539	0.6567	0.3753
Val_loss	1.5847	1.1745	1.4064	1.0946

모델 개발

모델의 정확도가 높은 쪽은?



3. 음식량 추정 모델

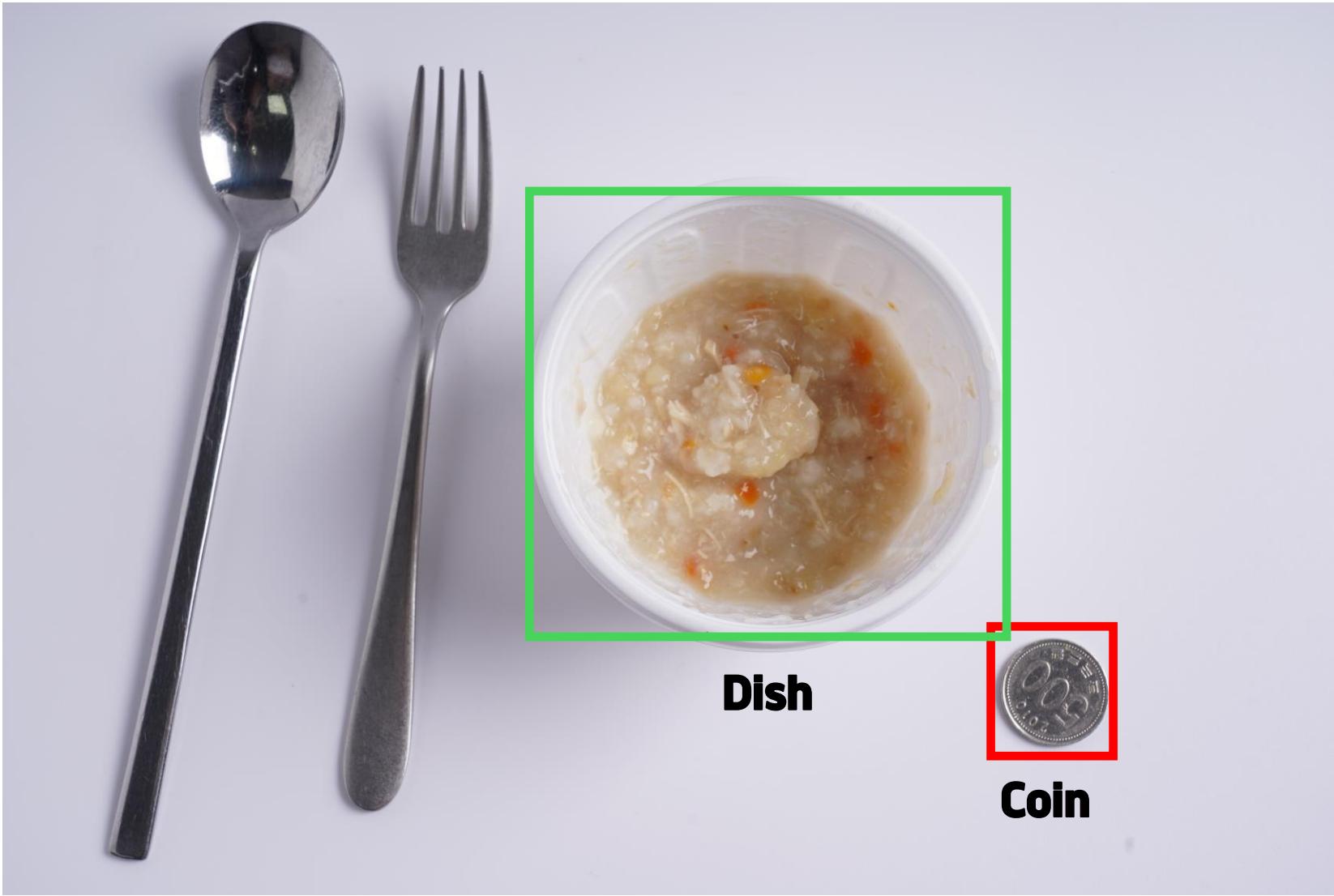
이미지 속 음식의
양을 1인분 기준으로
몇%에 해당하는지 판단

큰 차이는 없었으므로

Q1~Q5의 5단계로 추정하기로 결정

모델명	Q1~Q5 (1)	Q1~Q5 (2)	Q1, Q2+Q3, Q4+Q5 (1)	Q1, Q2+Q3, Q4+Q5 (2)
4,3)	(224,224,3)			
ynb	model8.ipynb			
ck	crop_black			
	16600			
	100			
전이학습	수제 모델			
Total params	223.02MB	223.08MB	222.77MB	222.79MB
Trainale params	222.47MB	222.50MB	222.22MB	222.24MB
Val_Accuracy	0.5636	0.6539	0.6567	0.3753
Val_loss	1.5847	1.1745	1.4064	1.0946

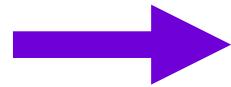
양추정 자체 모델 이미지 전처리



양추정 자체 모델 이미지 전처리



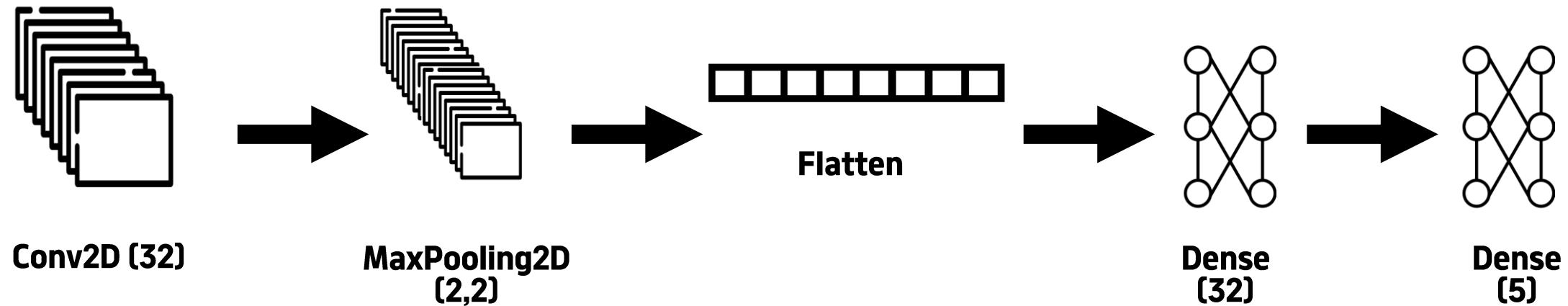
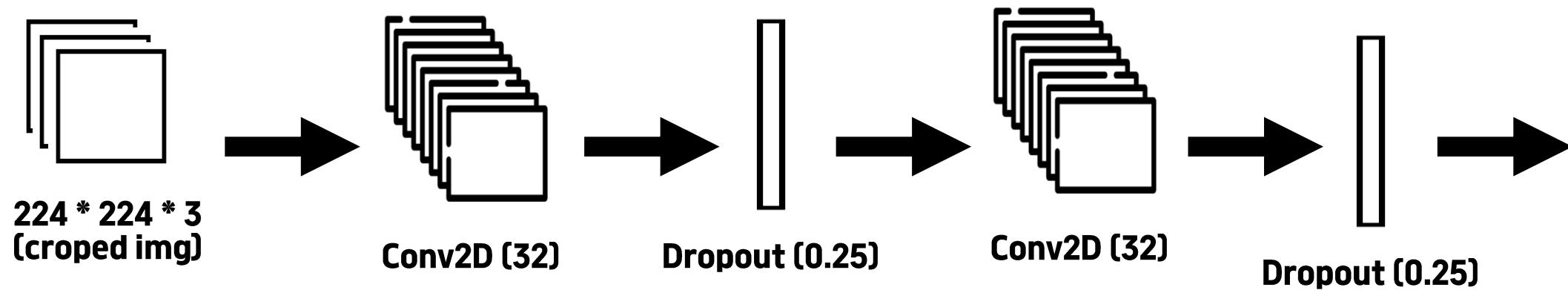
동전과 음식 사진을
단순 결합



resized
(224,224,3)

음식량 추정 모델 1(자체 모델)

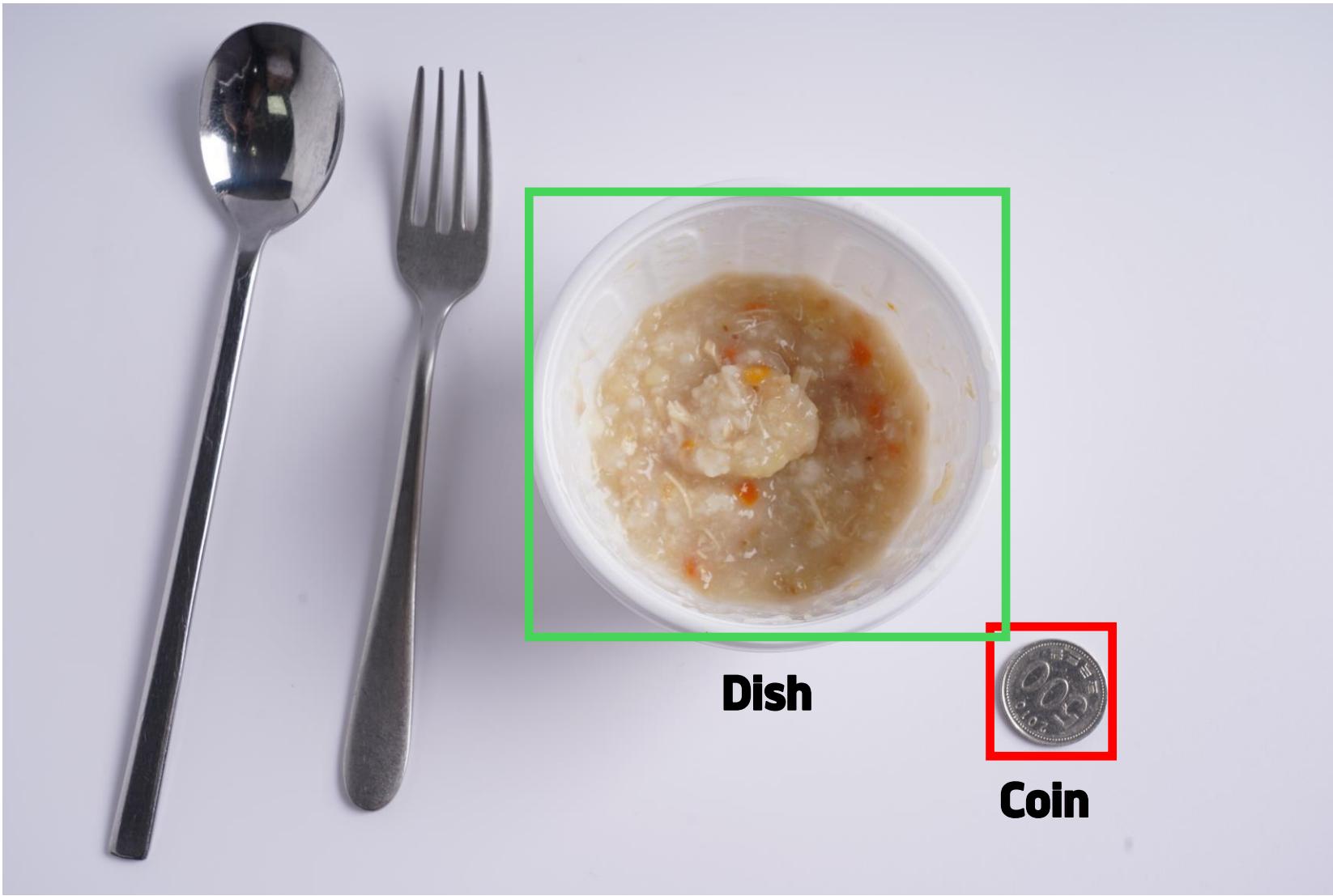
모델 개발



2. 음식량 추정 자체 모델

모델번호							7
Input Img	(224, 224, 3)						(224, 224, 3)
Learning File	model6.ipynb						model6.ipynb
Preprocess	crop_black						crop_black
Epochs							1000
Count Img							16600
Total params	45.62MB	46.48MB	46.48MB	46.48MB	46.48MB	93.04MB	94.61MB
Trainale para ms	45.62MB	46.48MB	46.48MB	46.48MB	46.48MB	93.04MB	94.61MB
Val_Accuracy	0.1943	0.1940	0.2407	0.1943	0.2569	0.1943	0.1975
Val_loss	1.6097	1.6097	3.7221	1.6097	1.6518	1.6097	1.6096

양추정 전이학습 모델 이미지 전처리



양추정 전이학습 모델 이미지 전처리

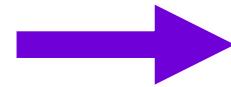


$$\div \quad =$$
A close-up photograph of a coin, specifically a 20-cent euro coin, centered within a red rectangular bounding box. The background is blurred.

동전 대비 음식의
크기 비율 (Ratio)

동전과 음식의 Bounding Box
크기 비율을 하나의 상수로
Model에 input함

양추정 전이학습 모델 이미지 전처리

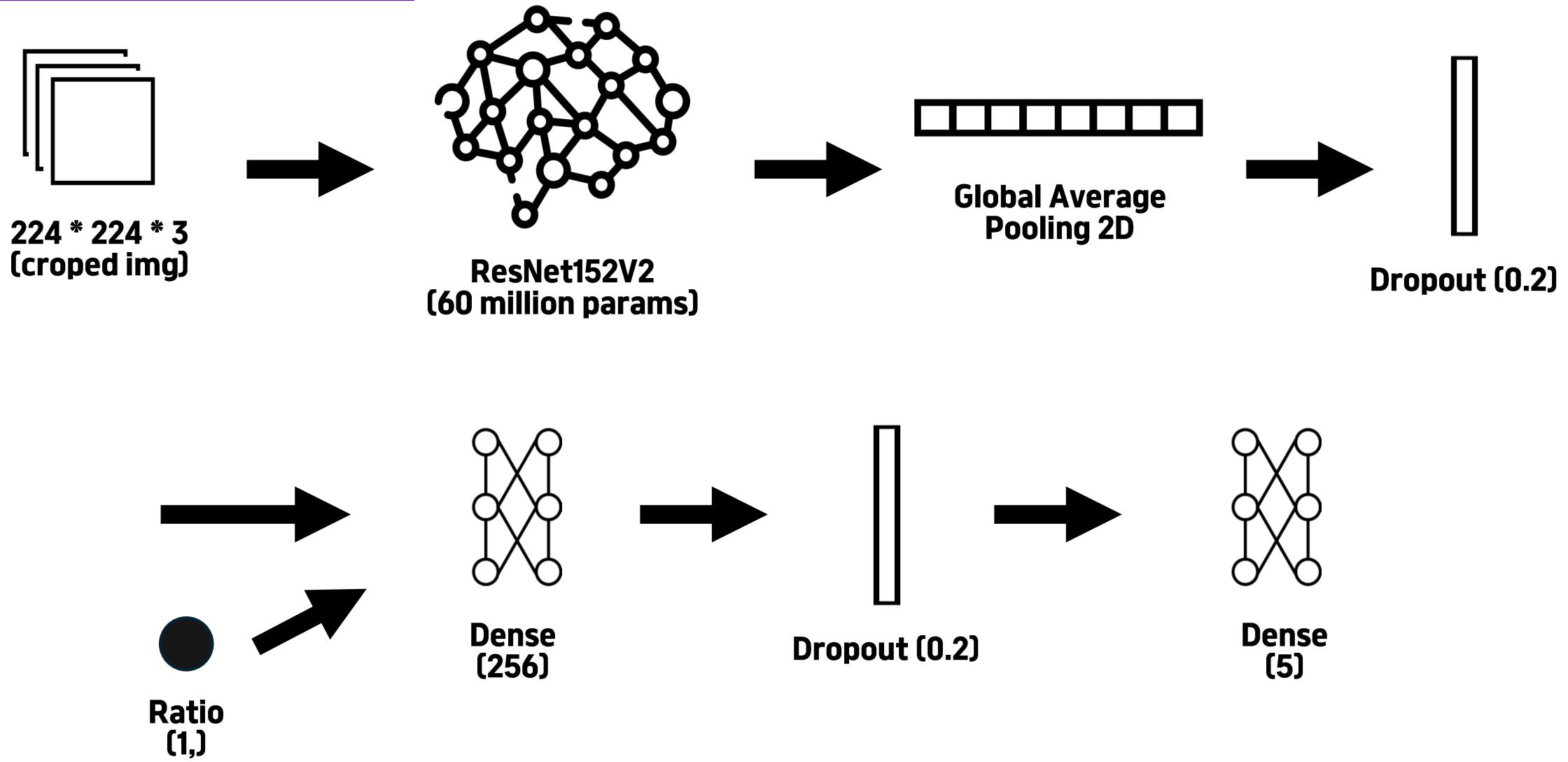


**resized
(224,224,3)**

음식사진만 따로 resize하여 모델에 input.
즉, input은 이미지 1장과 Ratio 상수 1개로 구성됨

음식량 추정 모델 2 (ResNet 전이학습)

모델 개발



2. 음식량 추정 전이학습 모델

3번 모델 검증 정확도

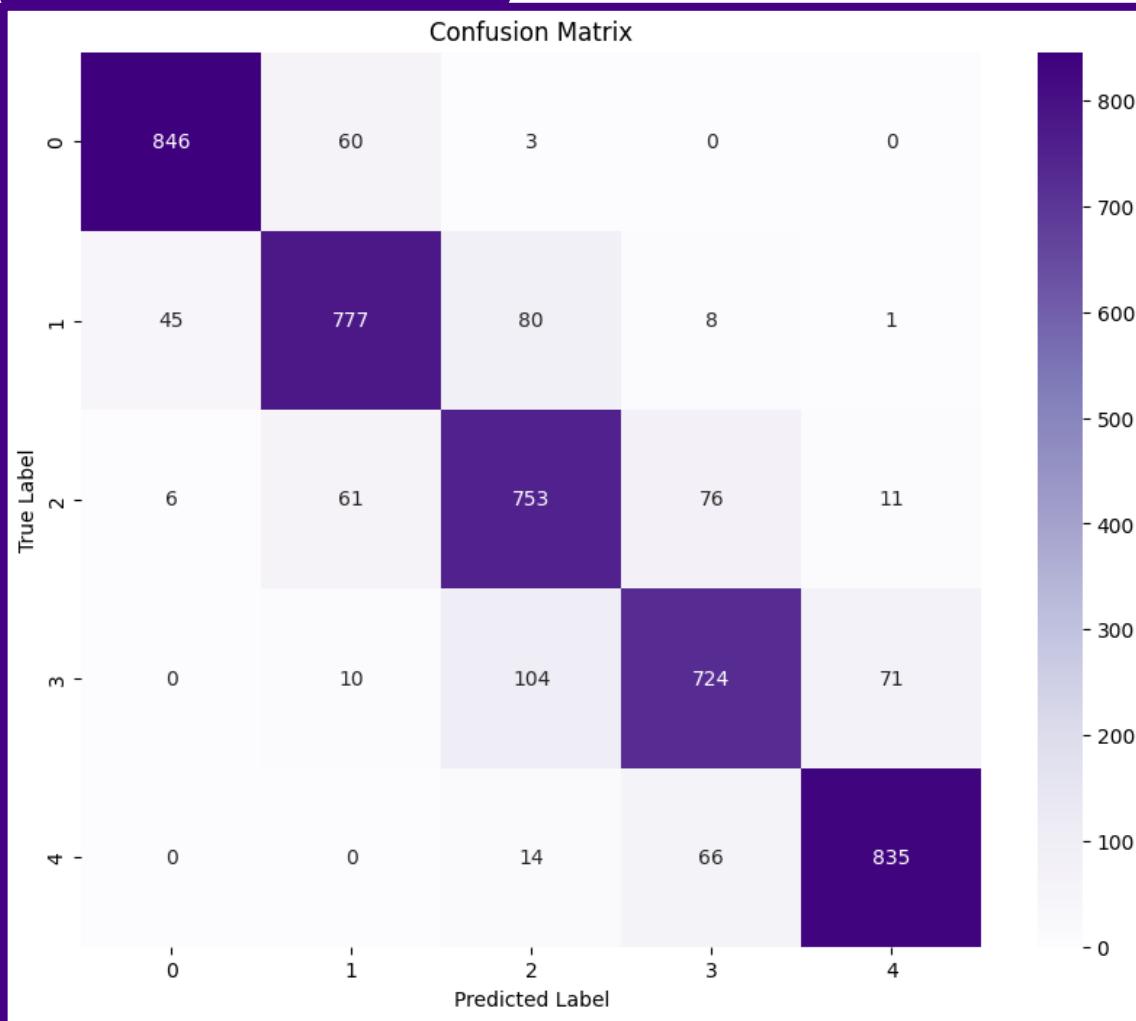
86% 달성

해당 모델을 베이스로
최종모델 학습!

모델번호							7
Input Img	(224, 224, 3)					24, 3)	(224, 224, 3)
Keras Model	m_0.85_keras					_0.85_keras	_0.85_keras
Preprocess	crop					Ratio	crop&Ratio
Epochs						0	100
Count Img	18259	18259	18259	18259	18259	18259	18259
특이사항	이미지 수 통일	-	데이터 stratify 시도, 기준을 val_acc로 변경	label을 연속형으로 취급	모델명에 val_loss 추가 val acc_val loss	-	-
Val_Accuracy	0.3979	0.4116	0.8590	0.3567 (mae)	0.8525	0.8467	0.8542
Val_loss	3.3698	4.1891	0.5010	0.2157	0.4844	0.4512	0.5944

2. 음식량 추정 전이학습 모델

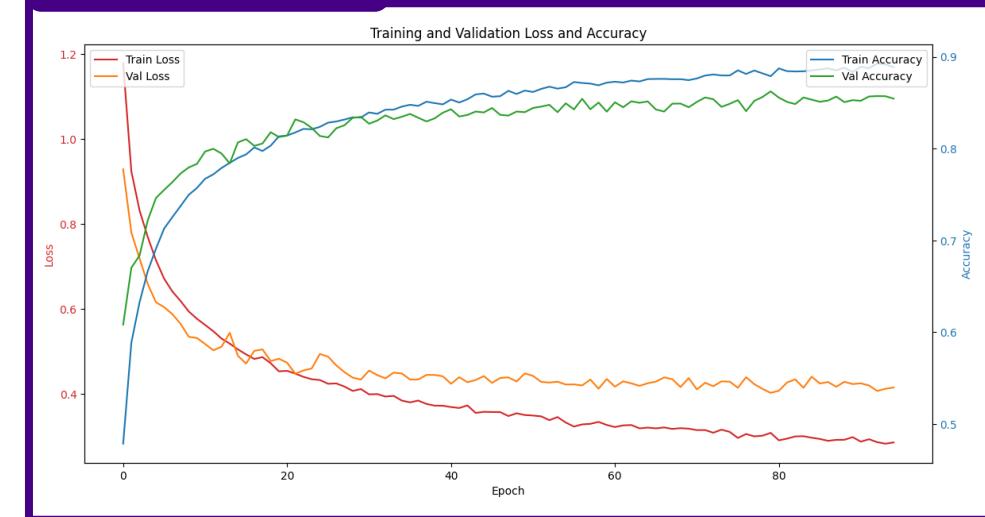
Confusion Metrix



Classification Report

	precision	recall	f1-score	support
Class 0	0.94	0.93	0.94	909
Class 1	0.86	0.85	0.85	911
Class 2	0.79	0.83	0.81	907
Class 3	0.83	0.80	0.81	909
Class 4	0.91	0.91	0.91	915
accuracy			0.86	4551
macro avg	0.87	0.86	0.86	4551
weighted avg	0.87	0.86	0.86	4551

Train History



On:ly

DIET PLANNER

ON:LY FOR YOU



ARE



칼로리 일일 섭취

권장량 제공

탄단지 일일 섭취

권장량 제공



건강을 고려한

적정 다이어트

기간 제공



AMERA



카메라로 촬영한 이미지에서
음식명, 칼로리, 탄단지
정보 추론 및 제공



이미지를 촬영하는 것 만으로도
그날의 식단과 영양정보를
기록, 관리 가능



음식과 동전을 함께 촬영하는 경우

음식의 양을 추정하여

탄단지, 칼로리 계산에 활용



동전이 함께 촬영되지 않은 경우

정량 1인분 기준으로

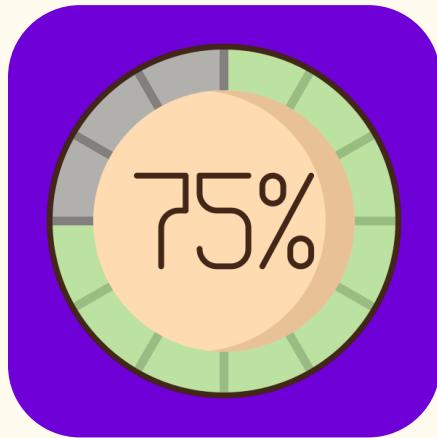
탄단지, 칼로리 정보 제공



CALENDAR

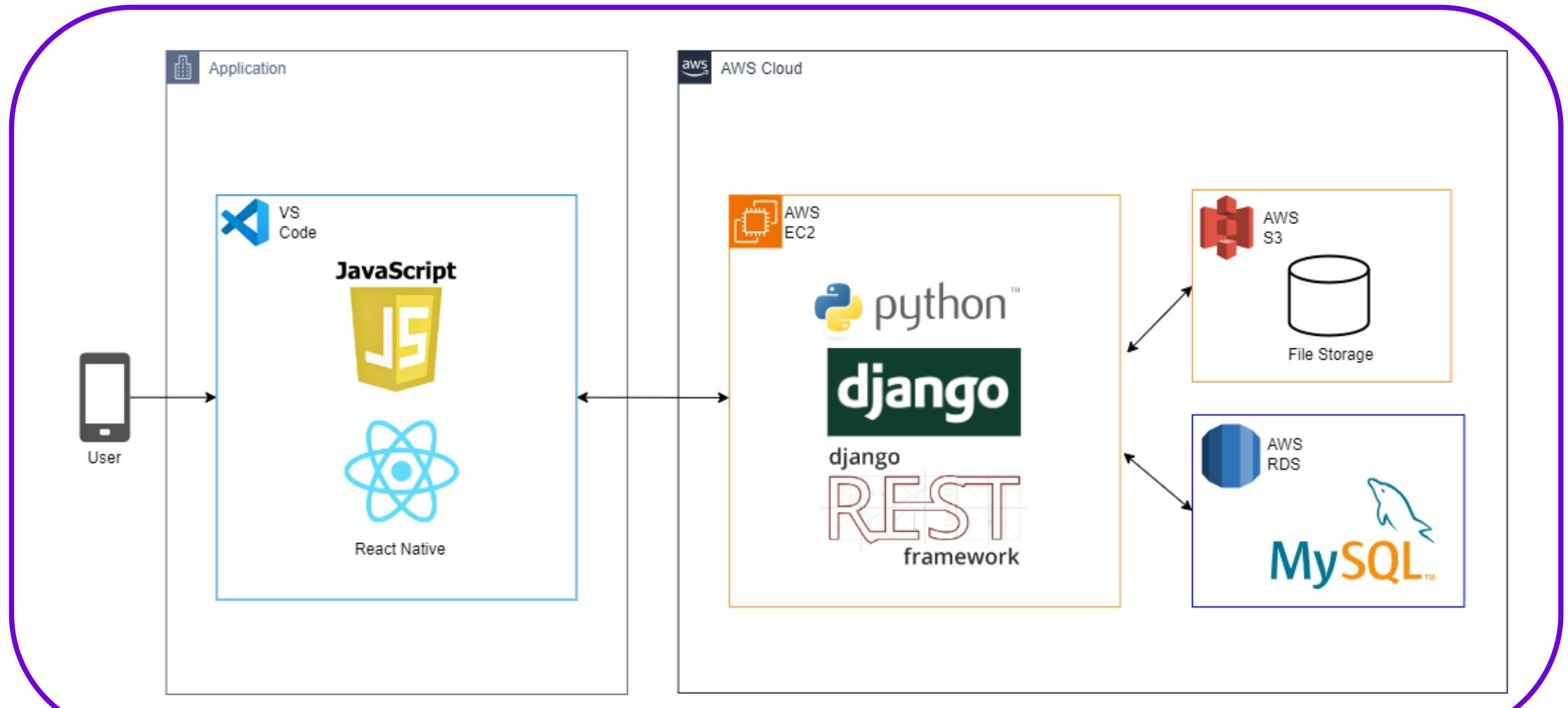


기록된 식단을
날짜별로 관리
날짜 별 다이어트
진행 상황 제공



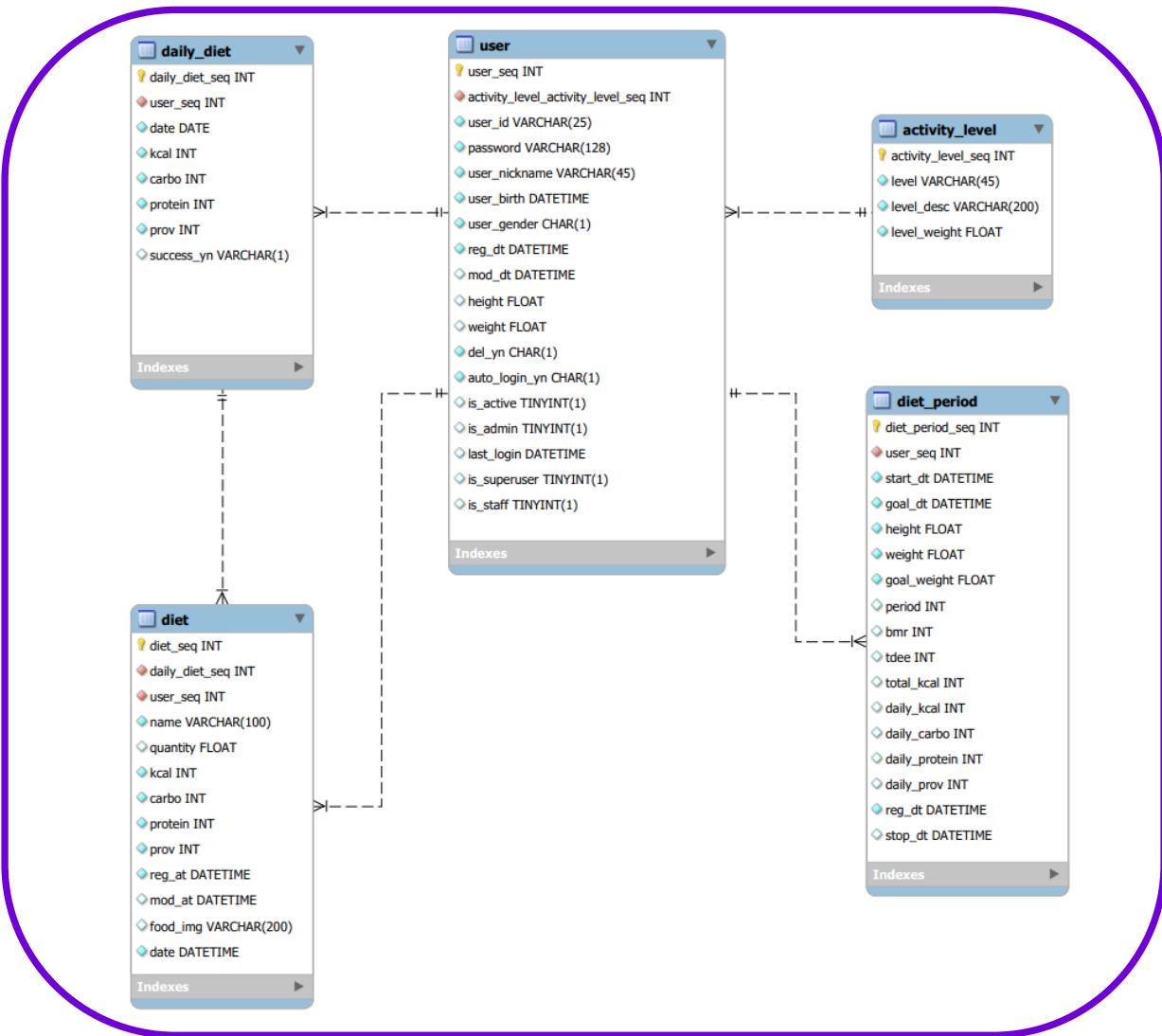
날짜 별 칼로리
섭취량을 기준으로
전체 다이어트
진행 상황 제공

System Architecture

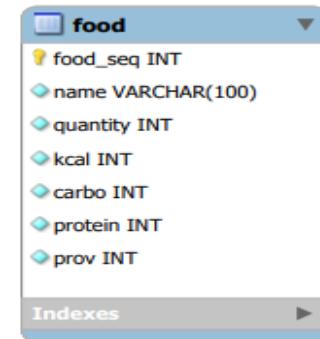


ERD 구성

어플리케이션 DB



모델관련 DB



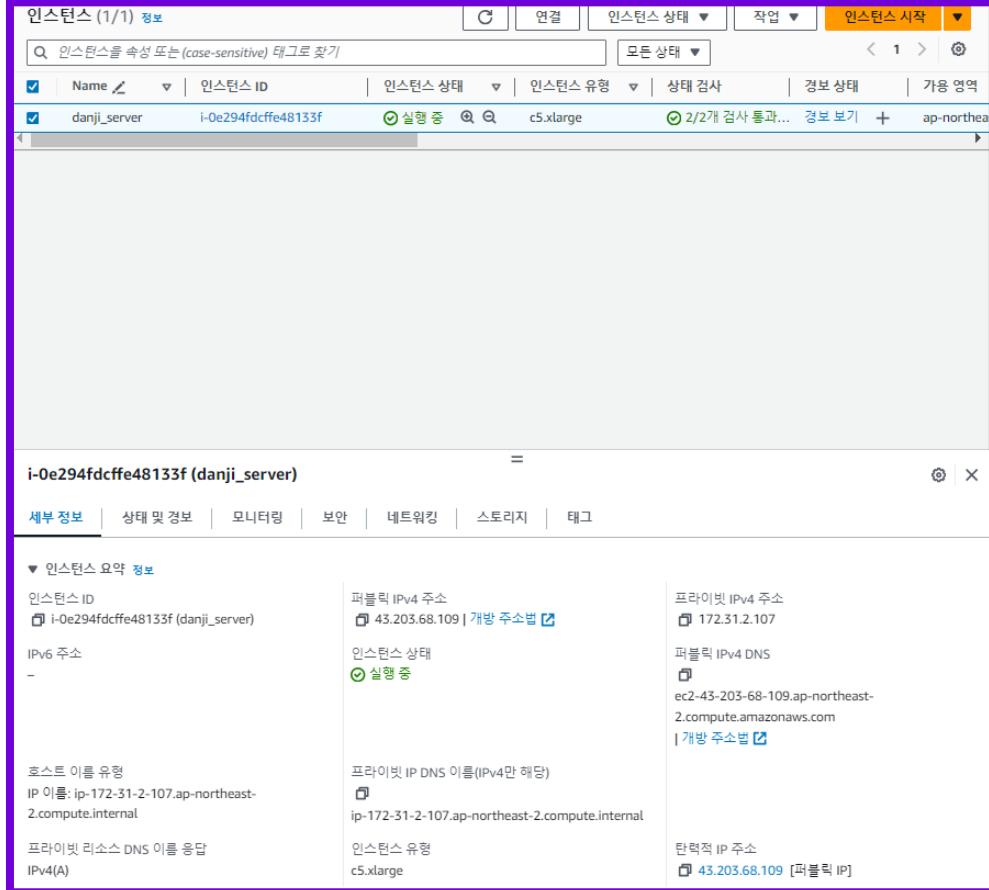
Backend 개발 환경

Back 개발환경 (puTTy)

```
ubuntu@ip-172-31-2-107: ~/dangi_model/diet2/dangi
[03/Jul/2024 12:39:36] "POST /diet/record/image/ HTTP/1.1" 200 206
[3, 3, 3, 4]
1/1 [=====] - 0s 139ms/step
[03/Jul/2024 12:39:45] "POST /diet/record/image/ HTTP/1.1" 200 206
[3, 3, 3]
1/1 [=====] - 0s 132ms/step
[03/Jul/2024 12:39:57] "POST /diet/record/image/ HTTP/1.1" 200 204
[3, 3, 3]
1/1 [=====] - 0s 130ms/step
[03/Jul/2024 12:40:12] "POST /diet/record/image/ HTTP/1.1" 200 206
[3, 3]
1/1 [=====] - 0s 131ms/step
[03/Jul/2024 12:40:19] "POST /diet/record/image/ HTTP/1.1" 200 206
[3, 3, 3]
1/1 [=====] - 0s 131ms/step
[03/Jul/2024 12:40:26] "POST /diet/record/image/ HTTP/1.1" 200 207
[3, 3, 3, 3]
1/1 [=====] - 0s 133ms/step
[03/Jul/2024 12:40:31] "POST /diet/record/image/ HTTP/1.1" 200 232
[3, 3, 3]
1/1 [=====] - 0s 131ms/step
[03/Jul/2024 12:40:40] "POST /diet/record/image/ HTTP/1.1" 200 207
[03/Jul/2024 12:41:55] "GET /user/diet_info/ HTTP/1.1" 200 338
[03/Jul/2024 12:41:55] "GET /diet/daily_meals/by-date/?date=2024-07-03 HTTP/1.1"
200 126
[03/Jul/2024 12:41:57] "GET /diet/daily_meals/ HTTP/1.1" 200 1566
Not Found: /diet/meals/by-date/
[03/Jul/2024 12:41:58] "GET /diet/meals/by-date/?date=2024-07-03T00:00:00Z HTTP/
1.1" 404 65
[03/Jul/2024 12:42:33] "POST /user/login/ HTTP/1.1" 200 518
[03/Jul/2024 12:42:33] "GET /user/diet_info/ HTTP/1.1" 200 338
[03/Jul/2024 12:42:33] "GET /diet/daily_meals/by-date/?date=2024-07-03 HTTP/1.1"
200 126
[03/Jul/2024 12:42:35] "GET /diet/daily_meals/ HTTP/1.1" 200 1566
Not Found: /diet/meals/by-date/
[03/Jul/2024 12:42:36] "GET /diet/meals/by-date/?date=2024-07-03T00:00:00Z HTTP/
1.1" 404 65
[3, 3, 3]
1/1 [=====] - 0s 133ms/step
[03/Jul/2024 12:43:47] "POST /diet/record/image/ HTTP/1.1" 200 213
[3, 3, 3]
1/1 [=====] - 0s 129ms/step
[03/Jul/2024 12:43:58] "POST /diet/record/image/ HTTP/1.1" 200 213
[3, 3, 3]
1/1 [=====] - 0s 133ms/step
[03/Jul/2024 12:44:13] "POST /diet/record/image/ HTTP/1.1" 200 213
[4, 3, 3]
1/1 [=====] - 0s 130ms/step
[03/Jul/2024 12:44:29] "POST /diet/record/image/ HTTP/1.1" 200 226
```

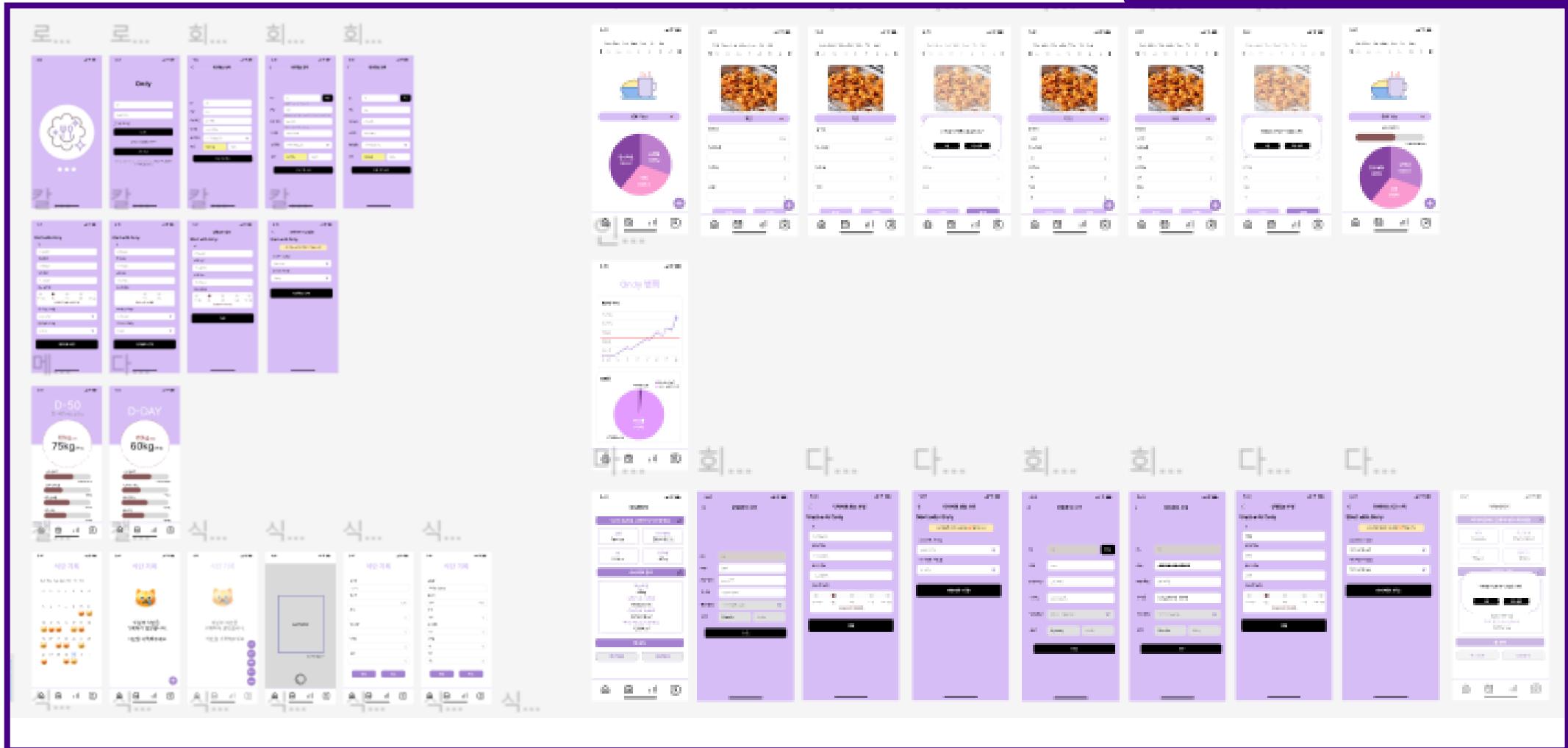


Back 개발환경 (EC2)



화면 설정

화면설계 wireframe



화면 설계

화면설계서(메인 홈 화면)

화면명	메인 홈 화면
참고 요구사항 번호	MAIN01_HOME
비고	



기능명세서	
1	다이어트 남은 기간을 보여준다. -기간이 종료가 되면 D+1..(증가일)을 보여준다.
2	[0>x] 메세지를 출력한다. [0<=x] 메세지를 출력한다.
3	user가 입력한 현재 / 목표 몸무게를 보여준다.
4~7	금일 섭취한 음식의 칼로리/탄수화물/단백질/지방 누적 정보 및 하루 권장량을 보여준다.
8	홈버튼 클릭시 현재 화면이 업데이트 된다.
9	캘린더 버튼 클릭시 MAIN02_CALENDAR 화면으로 이동한다.
10	그래프 버튼 클릭시 MAIN03_GRAPH 화면으로 이동한다.
11	마이페이지 버튼 클릭시 MAIN04_MYPAGE 화면으로 이동한다.

화면 설계

화면설계서(식단 목록 화면)

화면명	캘린더 화면 -> 식단 목록 화면_Default
참고 요구사항 번호	MAIN02_FOODLIST-02
비고	식단이 1개 이상 등록되어 있을 경우 발생되는 화면

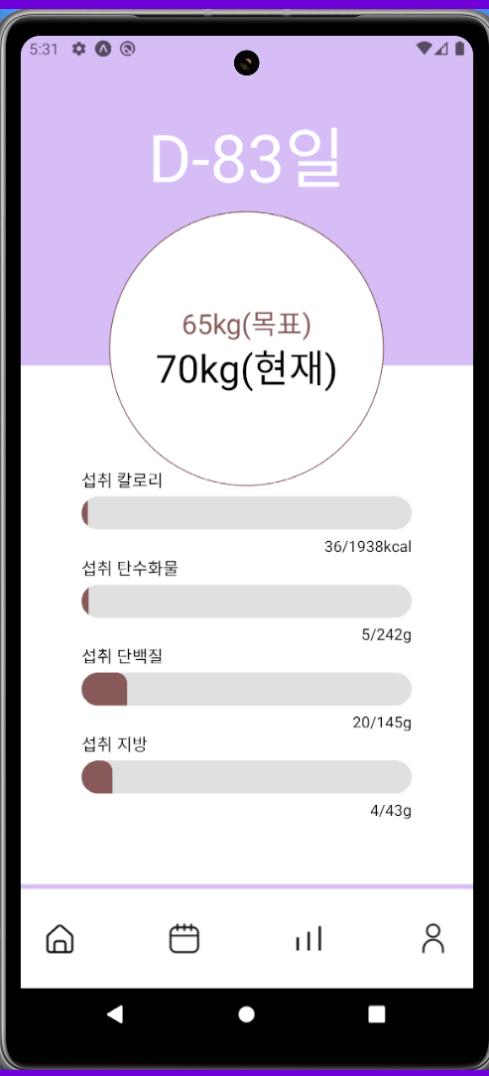


기능명세서

- 해당 일자 기준 주차 달력을 보여준다.
[식단이 있을 경우]
-원하는 일자 클릭하면 그 일자에 맞는 MAIN02_FOODLIST-02 화면을 보여준다.
[식단이 없을 경우]
-MAIN02_FOODLIST-01 화면으로 이동한다.
- user가 기록한 사진을 보여준다.
-text로 등록한 경우 디폴트 사진으로 보여준다.
- 하위 버튼 클릭시 전체~메뉴명을 보여준다.
-전체 메뉴가 보여지고, 메뉴 선택시 MAIN02_FOODLIST-02_SELECT 화면처럼 보여준다.
- user가 선택한 메뉴의 칼로리 결과를 바 그래프로 보여준다.
- user가 선택한 메뉴의 탄단지 결과를 원 그래프로 보여준다.
- +버튼 클릭시 MAIN02_FOODLIST-01화면처럼 보여준다.
- 홈버튼 클릭시 MAIN01_HOME 화면으로 이동한다.
- 캘린더 버튼 클릭시 MAIN02_CALENDAR 화면으로 이동한다.
- 그래프 버튼 클릭시 MAIN03_GRAPH 화면으로 이동한다.
- 마이페이지 버튼 클릭시 MAIN04_MYPAGE 화면으로 이동한다.

APP 개발 환경

APP 개발환경 (메인 페이지)



The mobile application displays a large purple header with the text "D-83일". Below it is a white circle containing the text "65kg(목표)" and "70kg(현재)". The main content area shows four progress bars for nutritional goals:

- 섭취 칼로리: 36/1938kcal
- 섭취 단수화물: 5/242g
- 섭취 단백질: 20/145g
- 섭취 지방: 4/43g

The application interface includes a navigation bar at the bottom with icons for home, calendar, list, and profile.

Code Snippets:

```
JS MainPage.js M JS Api.js M JS MyModifyPage.js M ... JS FoodList.js M X JS FoodListModify.js
```

```
JS MyModifyPage.js > ...
1 import React, { useState, useEffect } from
'react';
2 import { TextInput, View, Text,
TouchableOpacity, Alert } from
'react-native';
3 import { MyModifyPagestyles } from './
styles/MyModifyPagestyles';
4 import { SafeAreaView, SafeAreaProvider } from
'react-native-safe-area-context';
5 import { mydietinfo, updateUser } from './
Api'; // 기존 정보 불러오기 및 업데이트 함수
6 //as
7 const formatDate = (dateString) => {
8   const options = { year: 'numeric',
month: 'long', day: 'numeric' };
9   const date = new Date(dateString);
10  return date.toLocaleDateString('ko-KR',
options);
11 }
12
13 export default function MyModify({
navigation }) {
14   const [modID, setmodID] = useState("");
15   const [current_password,
setCurrent_password] = useState("");
16   const [current_password_check,
setCurrent_password_check] = useState
("");
17   const [new_nickname, setnew_nickname]
= useState("");
18   const [birth, setBirth] = useState
("") // 생년월일을 무자역으로 저장
}
16 const FoodList = () => {
522   <Text style={styles fabText}>카메라</Text>
523   <TouchableOpacity style={styles fabButton}>
524     onPress={() => handleOptionSelect('gallery')}>
525       <Text style={styles fabText}>갤러리</Text>
526       </TouchableOpacity>
527     <TouchableOpacity style={styles fabButton}>
528       onPress={() => handleOptionSelect('text')}>
529         <Text style={styles fabText}>텍스트</Text>
530         </TouchableOpacity>
531       <TouchableOpacity style={styles closeButton}>
532         onPress={toggleOptions}>
533           <Text style={styles closeButtonText}> - </Text>
534           </TouchableOpacity>
535     ) : (
536       <TouchableOpacity style={styles addButton} onPress=
{toggleOptions}>
537         <Text style={styles addButtonText}>+</Text>
538         </TouchableOpacity>
539     )
540     <View style={styles.buttonContainer}>
541       <TouchableOpacity style={styles saveButton} onPress=
{handleEditPress}>
542         <Text style={styles saveButtonText}>수정</Text>
543         </TouchableOpacity>
544       <TouchableOpacity style={styles cancelButton} onPress=
{handleDeletePress}>
545         <Text style={styles cancelButtonText}>삭제</Text>
546         </TouchableOpacity>
547     )
548   )
549 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

LOG goalDate: 2024-09-24T15:00:00.000Z
LOG todayDate: 2024-07-04T05:31:51.747Z
LOG diffTime: 711888253
LOG D-day: 83

Ln 518. Col 17 Spaces: 2 UTF-8 CRLF {} JavaScript ⚡ Go Live ✨ Prettier

API 정의

API 명세서 (Notion)

FE	BE	도메인	담당자	HTTP 메서드	URL	model(table)	Aa 기능	활용
완료	완료	사용자	가연 류	POST	/user/member/	diet_period user	회원가입	
완료	완료	사용자	가연 류	GET	/user/member/<str:user_id>/	user	ID 중복 체크	
완료	완료	사용자	가연 류	GET	/user/member	user	회원정보 조회(MyPage viewer)	
완료	완료	사용자	가연 류	PUT	/user/member	user	회원정보 수정(닉네임)	
완료	완료	사용자	가연 류	POST	/user/login	user	로그인	
완료	완료	사용자	가연 류	POST	/user/logout	user	로그아웃	
완료	완료	사용자	가연 류	DELETE	/user/delete	user	회원탈퇴	
완료	완료	사용자	가연 류	GET	/user/diet_info	diet_period	다이어트 정보 조회(MyPage viewer)	
완료	완료	사용자	가연 류	PUT	/user/diet_info	diet_period user	다이어트 정보 수정(NewUser Signup)	다이어트 추천기간도 포함
완료	완료	다이어트	가연 류	POST	/diet/meals	daily_diet diet	식단 기록	
완료	완료	다이어트	가연 류	PUT	/diet/meals/<int:diet_pk>/	daily_diet diet	식단 수정	
완료	완료	다이어트	가연 류	DELETE	/diet/meals/<int:diet_pk>/	daily_diet diet	식단 삭제	
완료	완료	다이어트	가연 류	GET	/diet/meals/by-date/	diet	식단 목록 조회 [날짜별]	
완료	완료	다이어트	가연 류	GET	/diet/daily_meals/by-date/	daily_diet	일일 섭취량 조회	
완료	완료	다이어트	가연 류	GET	/diet/daily_meals/period/	daily_diet	지정기간 일일 섭취량 조회	인사이트 웹부분, 한달 캘린더 그릴때 사용하면 됨.
완료	완료	다이어트	가연 류	GET	/diet/daily_meals/	daily_diet	모든 일일 섭취량 조회	
완료	완료	다이어트	가연 류	GET	/diet/progress	daily_diet diet_period	다이어트 진행률 그래프	
완료	완료	다이어트	pass life	POST	/diet/record/image	S3 deep_learning food	갤러리 혹은 카메라 이미지 통한 식단 기록	
생략	생략	사용자	가연 류	GET	/user/diet_info/recommend_period/	diet_period	다이어트 추천 기간 계산	
생략	생략	사용자	가연 류	GET	/user/diet_height	user	다이어트정보 입력 여부 확인 [몸무게]	

API 정의

API 명세서 상세 [회원가입]

요청 헤더

- Content-Type : application/json

요청 본문

- 형식: JSON

요청 파라미터

필드명	타입	필수 여부	설명	유효성 검사
user_id	String	필수	사용자 ID	영어와 숫자만 가능, 4-12자 길이
userNickname	String	필수	사용자 닉네임	영어, 숫자, 한글만 가능, 2-15자 길이
password	String	필수	비밀번호	영어, 숫자, 특수문자를 최소 1개씩 포함, 8-15자 길이
passwordCheck	String	필수	비밀번호 확인	비밀번호와 일치해야 함
userBirth	DateTime	필수	생년월일	형식: YYYY-MM-DD HH:MM:SS
userGender	String	필수	성별	"M" 또는 "F"

요청 예시

```
{  
    "user_id": "fiansd0d5s",
```

API 동작

APP의 API 요청(메인 페이지)

server의 API 응답(메인 페이지)

```
218     class DailyDietByDateView(APIView):
219         permission_classes = [IsAuthenticated]
220
221         45 usages (45 dynamic) ▲ ryugayeon +1
222         def get(self, request):
223             date_str = request.query_params.get('date')
224             if not date_str:
225                 return Response( data: {'error': 'Date parameter is required'}, status=status.HTTP_400_BAD_REQUEST)
226
227             date_obj = parse_datetime(date_str)
228             if not date_obj:
229                 return Response( data: {'error': 'Invalid date format'}, status=status.HTTP_400_BAD_REQUEST)
230
231             # 특정 날짜와 일치하는 daily_diet 레코드 찾기
232             daily_diets = DailyDiet.objects.filter(date__date=date_obj.date()).order_by('-date')
233
234             if not daily_diets.exists():
235                 return Response( data: {'error': 'No daily_diets found for the specified date'}, status=status.HTTP_404_NOT_FOUND)
236
237             # 특정 날짜와 일치하는 daily_diet가 여러개인 경우
238             # 가장 최근의 daily_diet 레코드 가져오기
239             latest_daily_diet = daily_diets.first()
240             serializer = DailyDietSerializer(latest_daily_diet)
241             return Response(serializer.data, status=status.HTTP_200_OK)
```

```
//mainpage (user정보+현재까지 먹은 칼로리 정보)
export const mainInfo = async (date) => {
    try {
        token = await AsyncStorage.getItem('accessToken')

        const userInfoResponse = await apiClient.get('/user/diet_info/');
        console.log('UserInfoResponse:', userInfoResponse.data);

        console.log('Request Date:', date);
        const mealInfoResponse = await apiClient.get('/diet/daily_meals/by-date', { params: { date } });
        console.log('MealInfoResponse:', mealInfoResponse.data);

        let kcal = 0;
        let carbo = 0;
        let protein = 0;
        let prov = 0;

        if (!mealInfoResponse.data.error) {
            kcal = mealInfoResponse.data.kcal || 0;
            carbo = mealInfoResponse.data.carbo || 0;
            protein = mealInfoResponse.data.protein || 0;
            prov = mealInfoResponse.data.prov || 0;
        }

        const mainuserInfo = {
            goal_dt: userInfoResponse.data.goal_dt,
            weight: userInfoResponse.data.weight,
            goal_weight: userInfoResponse.data.goal_weight,
            kcal,
            carbo,
            protein,
            prov,
            daily_kcal: userInfoResponse.data.daily_kcal,
            daily_carbo: userInfoResponse.data.daily_carbo,
            daily_protein: userInfoResponse.data.daily_protein,
            daily_prov: userInfoResponse.data.daily_prov,
        };

        return mainuserInfo;
    } catch (error) {
        if (error.response && error.response.data.error === "No daily_diets found for the specified date") {
            const userInfoResponse = await apiClient.get('/user/diet_info/');
        }
    }
};
```

Directory Structure

Django (Server)

```
dangi
├── config
│   ├── asgi.py
│   ├── my_settings.py
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   └── __init__.py
├── diet
│   ├── img_DeepLearning
│   │   ├── img_FoodClassify.py
│   │   ├── img_FoodQuantity.py
│   │   ├── img_GetDBinfo.py
│   │   ├── img_Inference.py
│   │   ├── img_ModelPreloader.py
│   │   ├── img_ObjectDetect.py
│   │   ├── img_S3FileManagement.py
│   │   ├── makedummym.py
│   │   └── __init__.py
│   ├── migrations
│   │   ├── 0001_initial.py
│   │   ├── 0002_initial.py
│   │   ├── 0003_food_foodamount_alter_dailydiet_success_yn.py
│   │   ├── 0004_delete_foodamount_alter_food_name.py
│   │   └── __init__.py
├── models
│   ├── keras
│   │   ├── add_model13.hdf5
│   │   ├── model_0.86_0.40_saved.h5
│   │   └── newclass.csv
│   └── yolo
│       ├── obj.names
│       ├── yolov3_testing.cfg
│       └── yolov3_training_final.weights
└── user
    ├── migrations
    │   ├── 0001_initial.py
    │   ├── 0002_alter_user_options.py
    │   └── __init__.py
    ├── admin.py
    ├── apps.py
    ├── models.py
    ├── serializers.py
    ├── tests.py
    ├── views.py
    └── __init__.py
    └── fix_migrations.py
    └── manage.py
    └── requirements.txt
    └── test.py
```

Directory Structure

React Native (APP)

```
📦 Func
├── Add_Text.js
├── Api.js
├── App.js
├── Cal.js
├── CalendarScreen.js
├── config.js
├── dietInfo.js
├── FoodContext.js
├── FoodList.js
├── FoodListModify.js
├── Insight.js
├── loading.js
├── Login_typing.js
├── MainPage.js
├── ModifyPage.js
├── MyModifyPage.js
└── MyPage.js

📦 styles
├── Appstyles.js
├── dietInfostyles.js
├── InsightStyles.js
├── loadingstyles.js
├── Login_typing_styles.js
├── MainPagedestyles.js
├── ModifyPagedestyles.js
├── MyModifyPagedestyles.js
├── Mypagedestyles.js
├── SignupFormstyles.js
├── styles.js
├── styles_calendar.js
├── styles_foodlist.js
├── styles_modify.js
└── styles_text.js

📦 assets
├── adaptive-icon.png
├── angry_cat.png
├── app_icon.png
├── calendar.png
├── cry_cat.png
├── default_food.png
├── favicon.png
├── heart_cat.png
├── home.png
├── icon.png
├── insight.png
├── loading.png
├── modify.png
├── profile.png
└── splash.png
```

04

결과물 소개

By Danzi

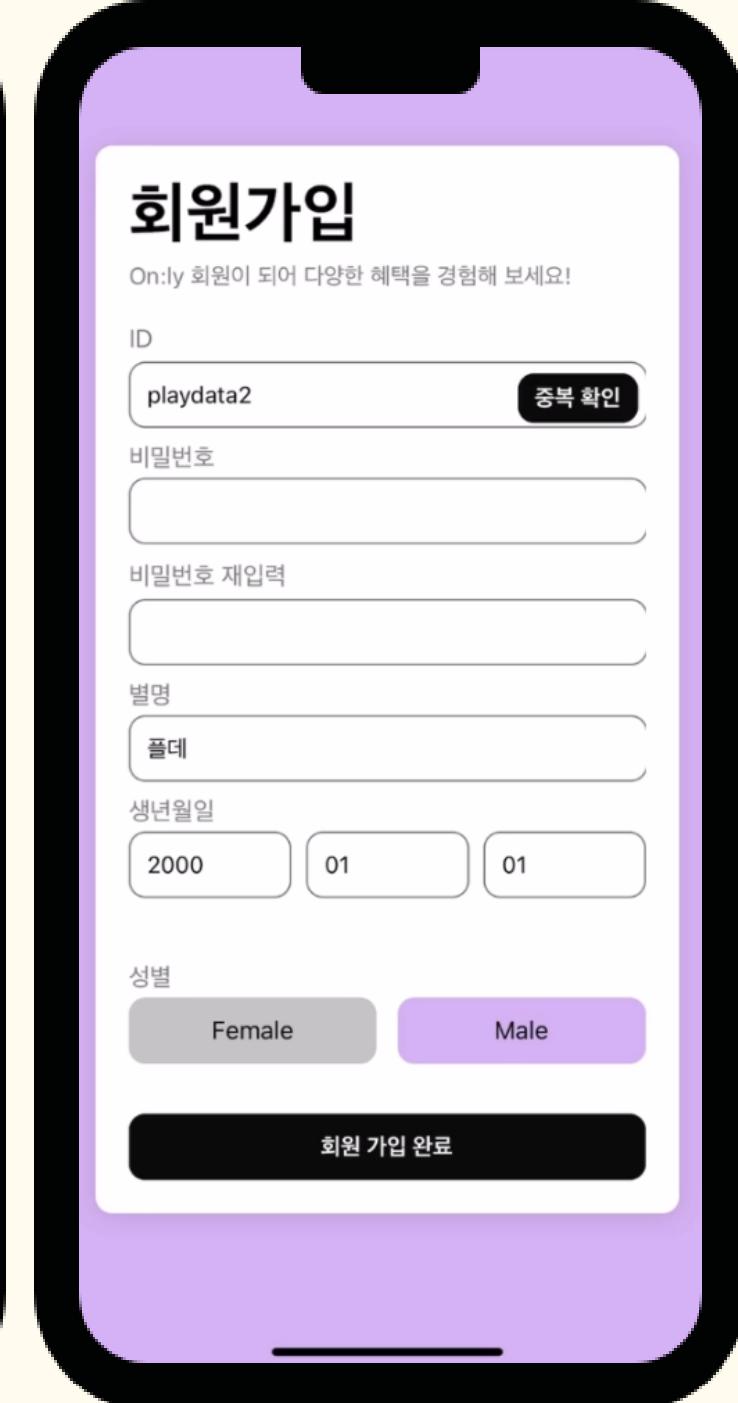
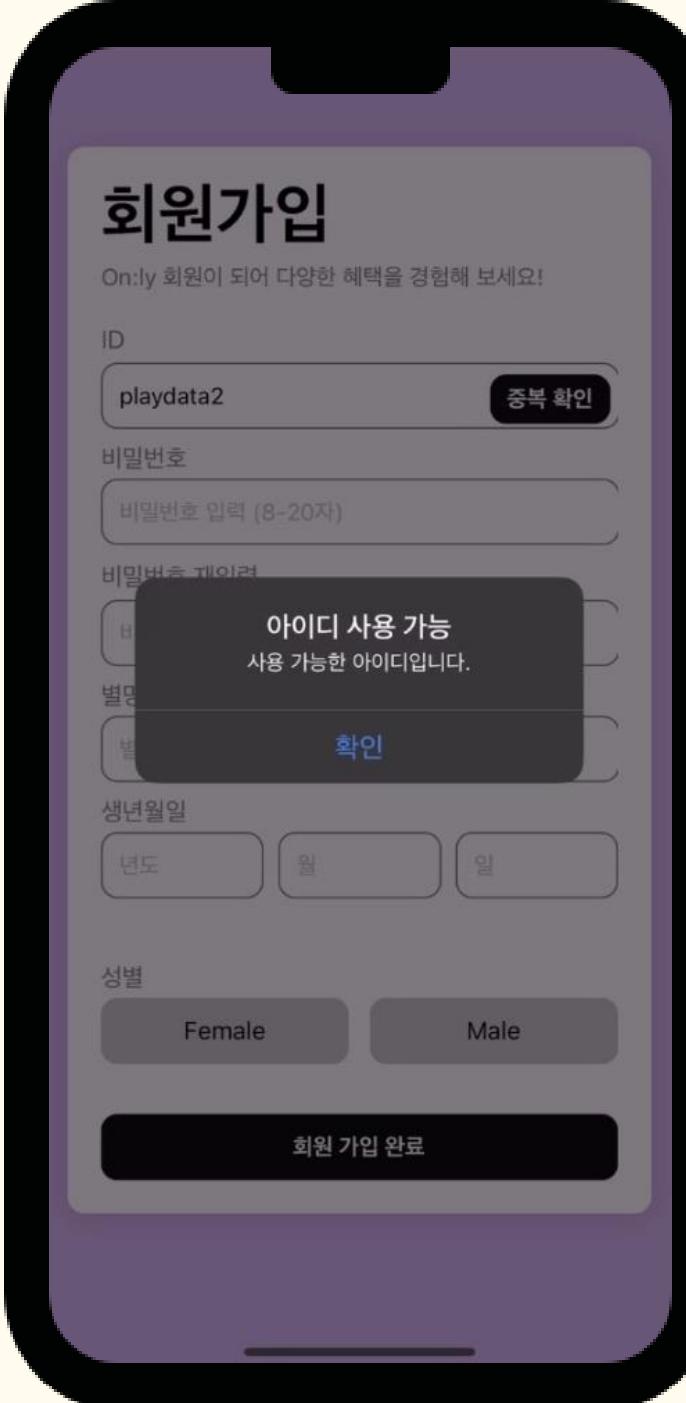


On:ly

DIET PLANNER
ON:LY FOR YOU

기능 소개

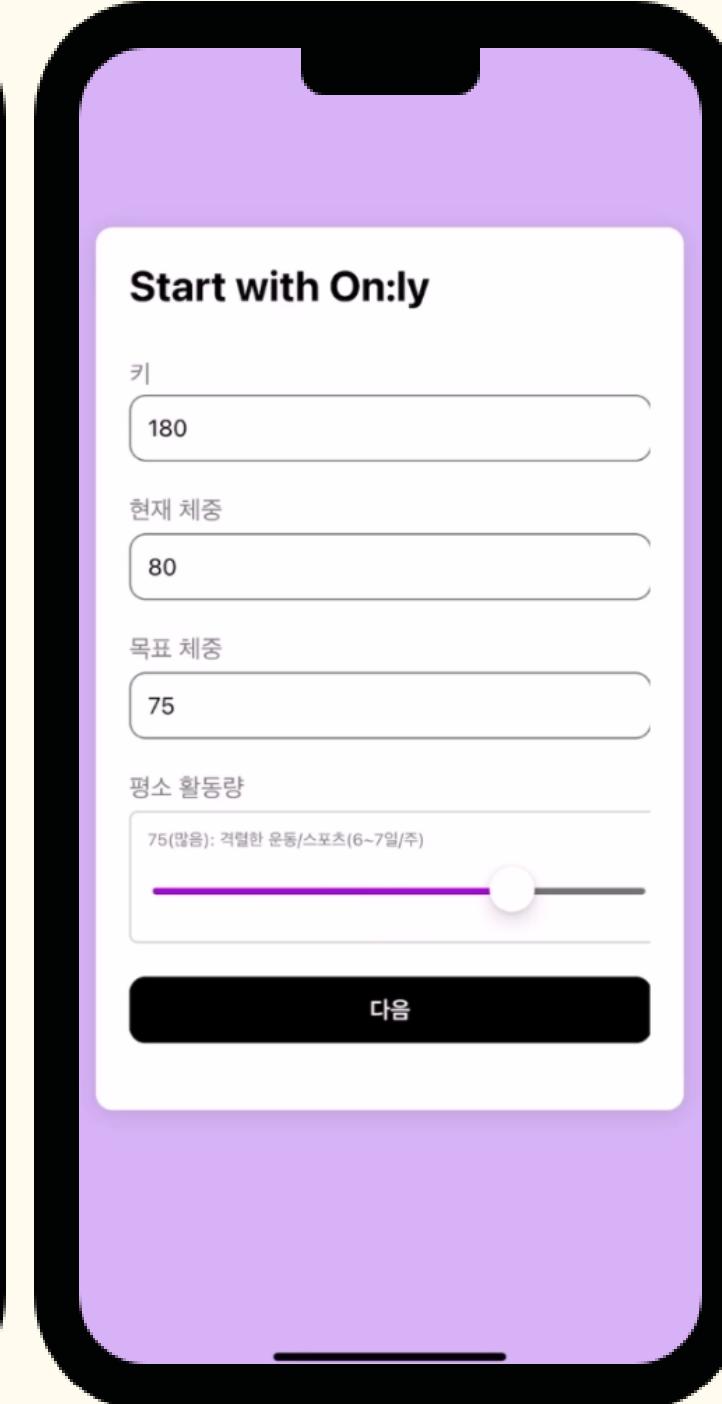
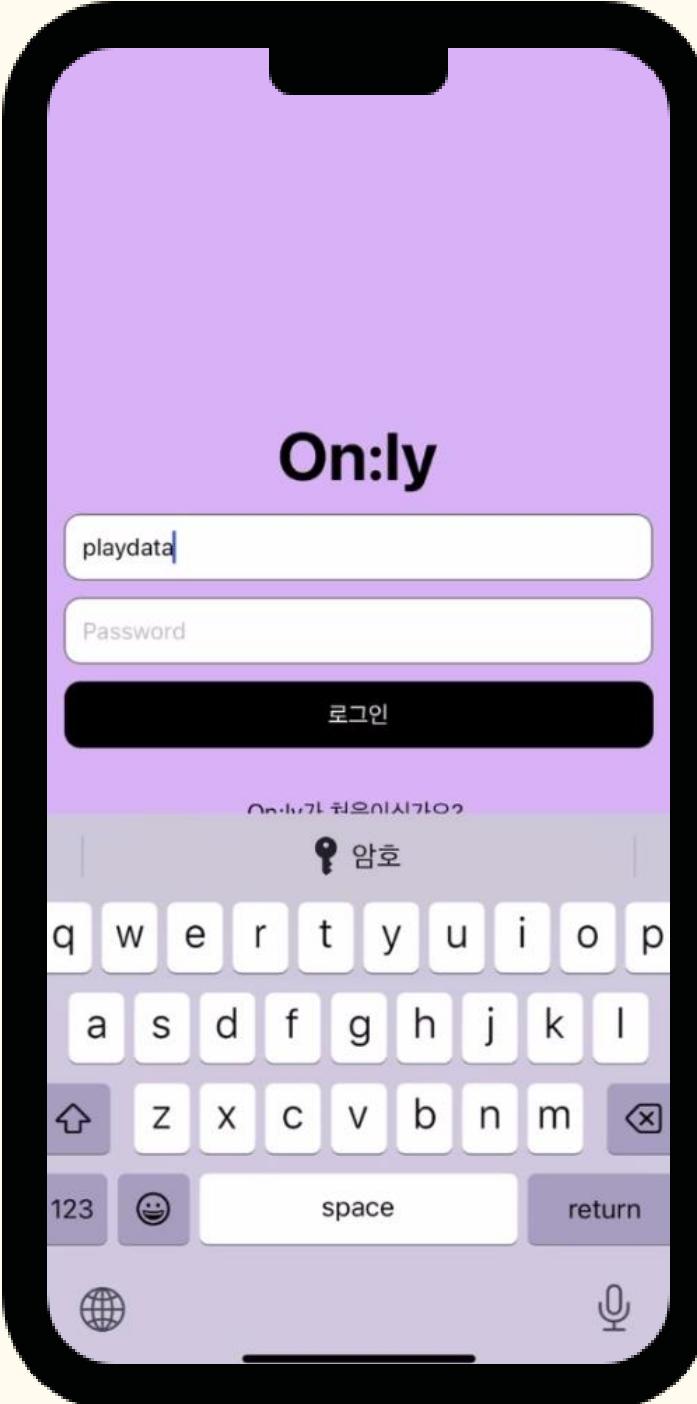
회원가입



기능 소개

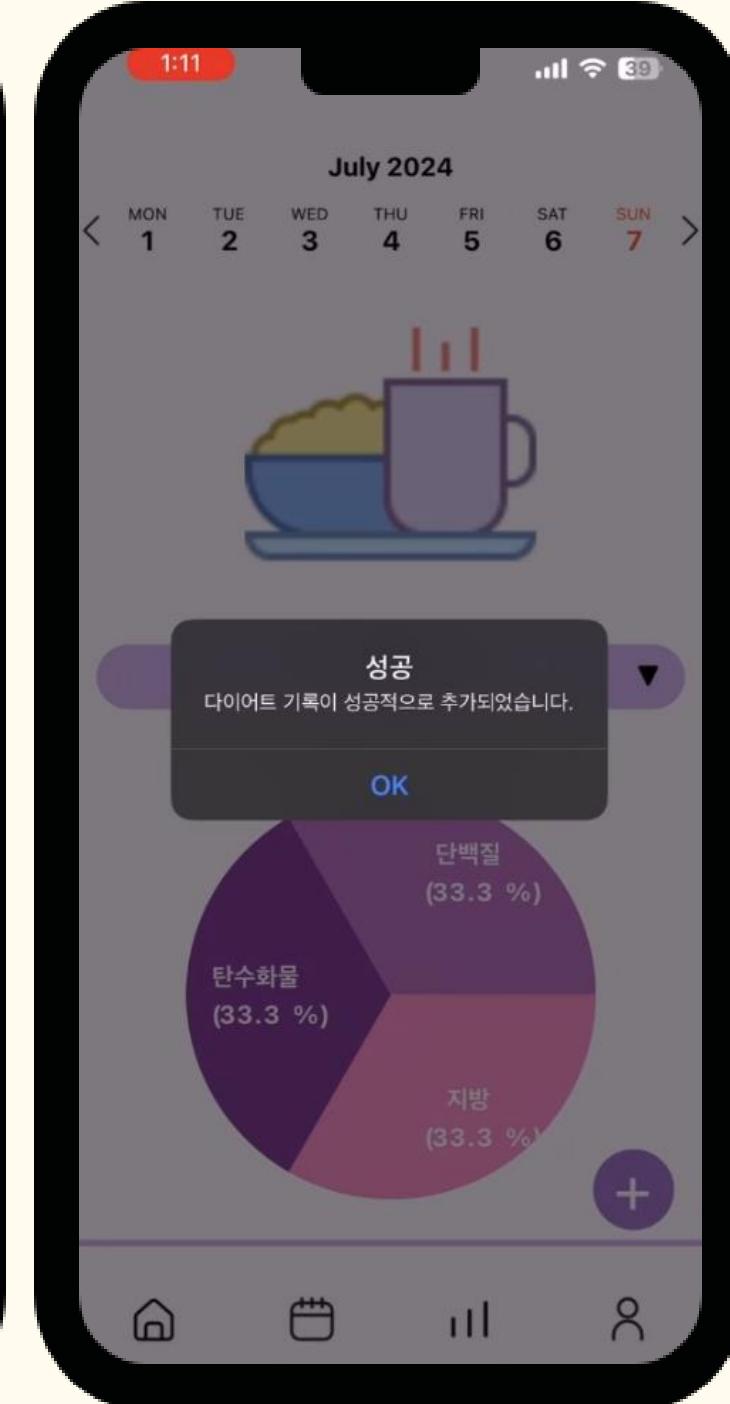
로그인

다이어트 정보 입력



기능 소개

식단정보 입력 (텍스트)



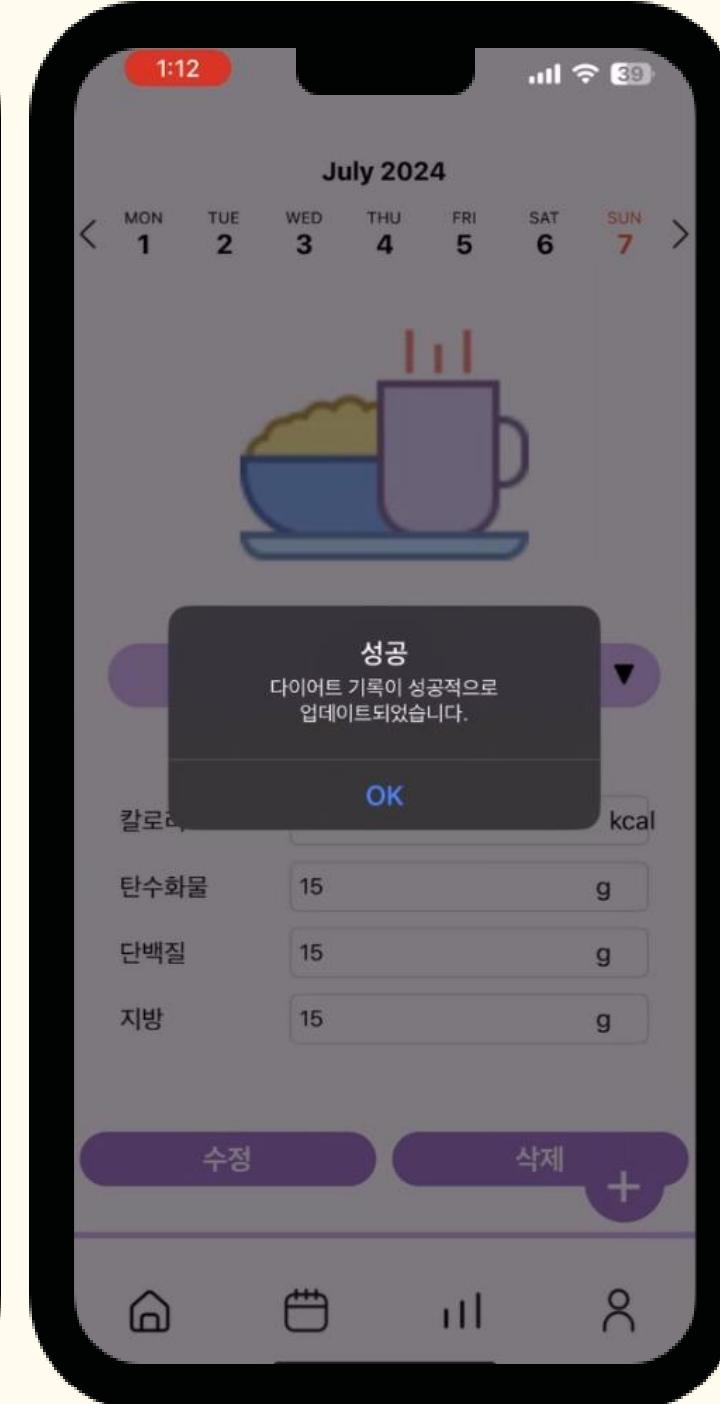
기능 소개

식단정보 입력 (갤러리)



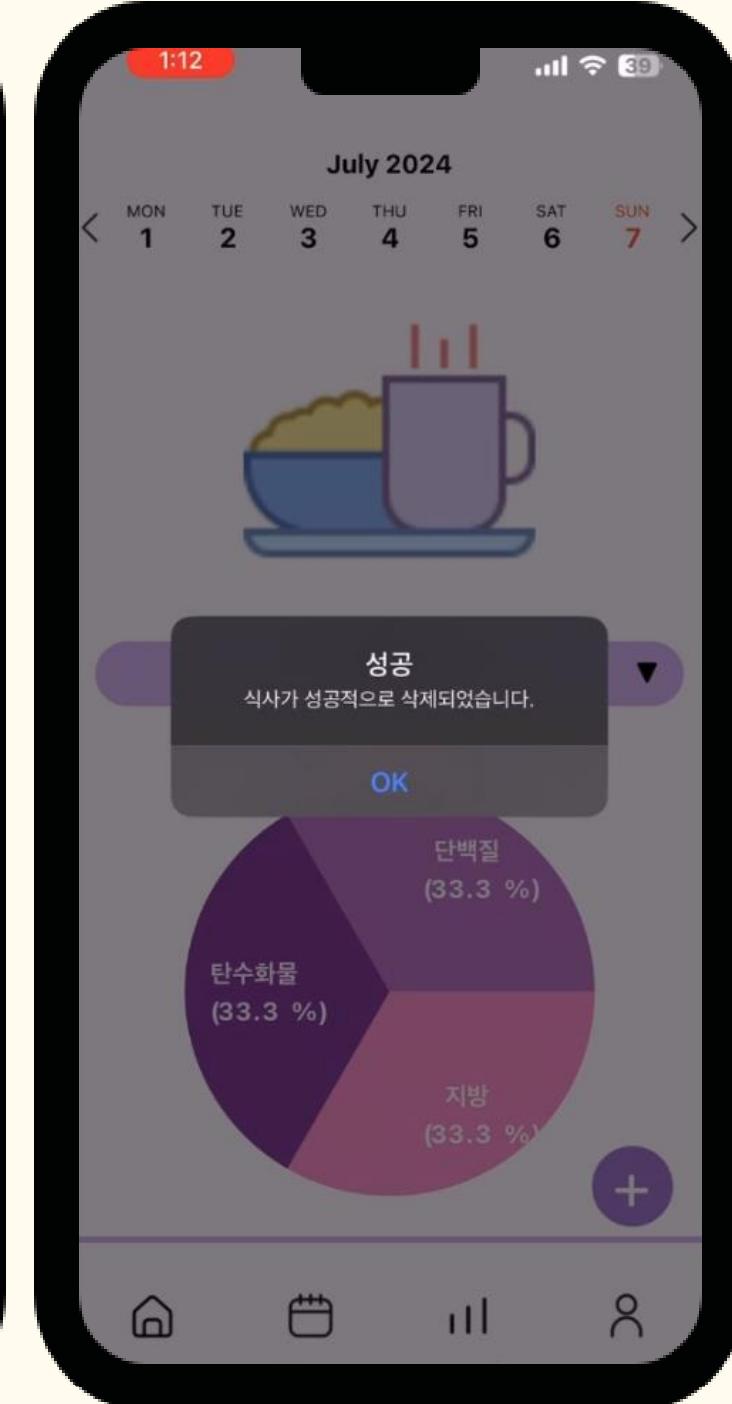
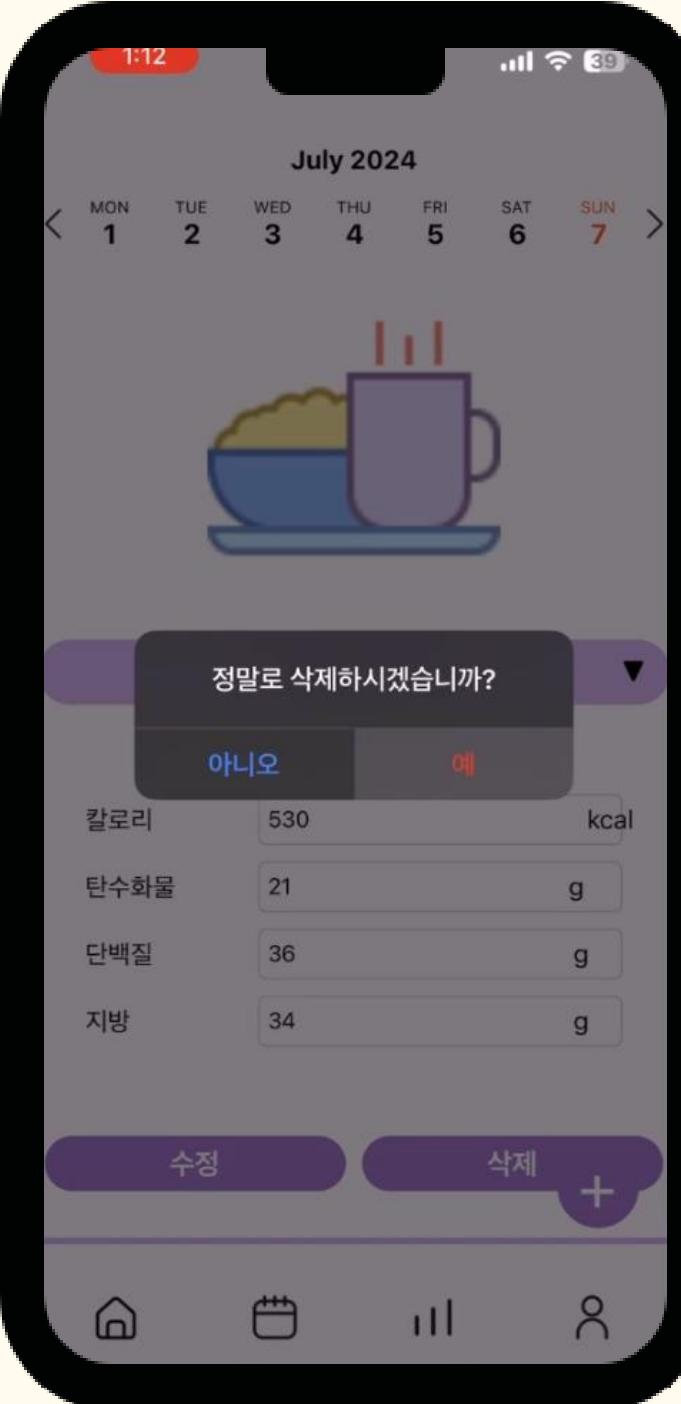
기능 소개

식단정보 설정



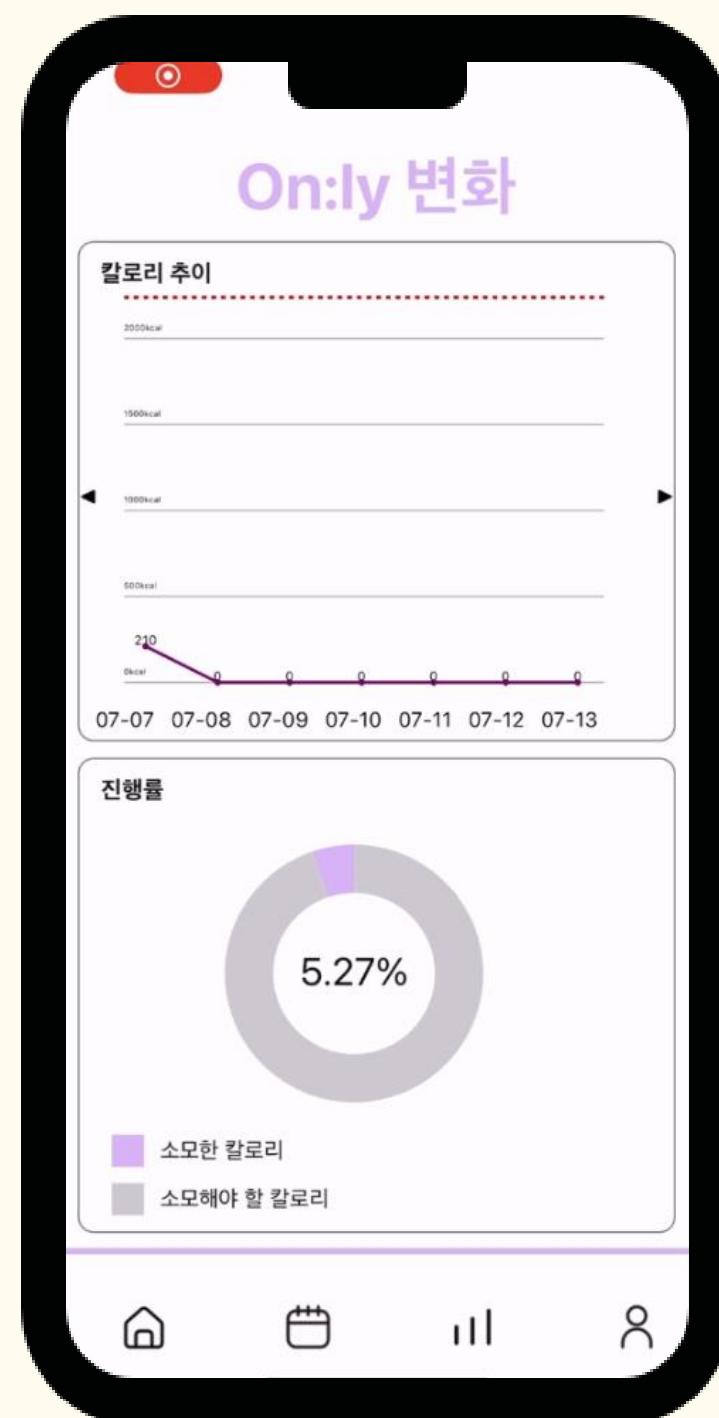
기능 소개

식단정보 삭제



기능 소개

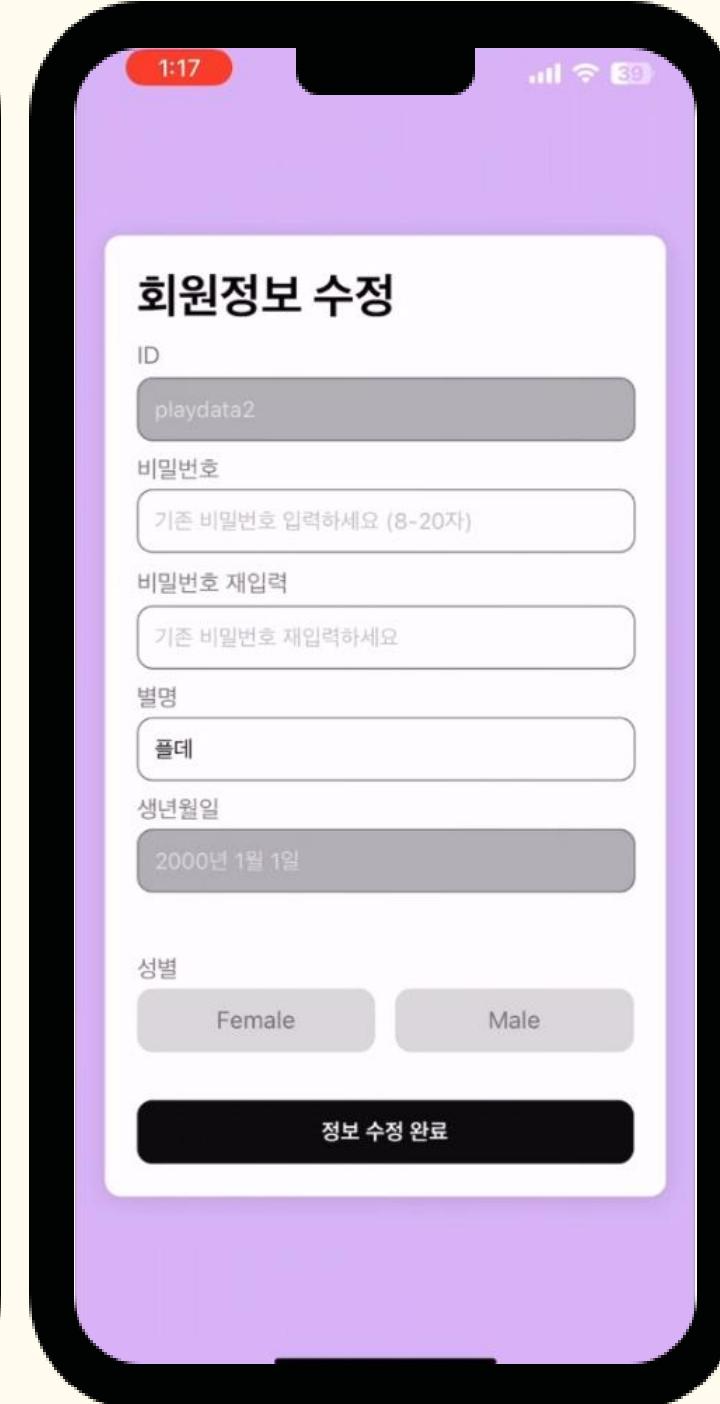
다이어트 진행상황



기능 소개

회원정보 설정

로그아웃



05

〔 QnA 〕

By Danzi



On:ly

Thank you!